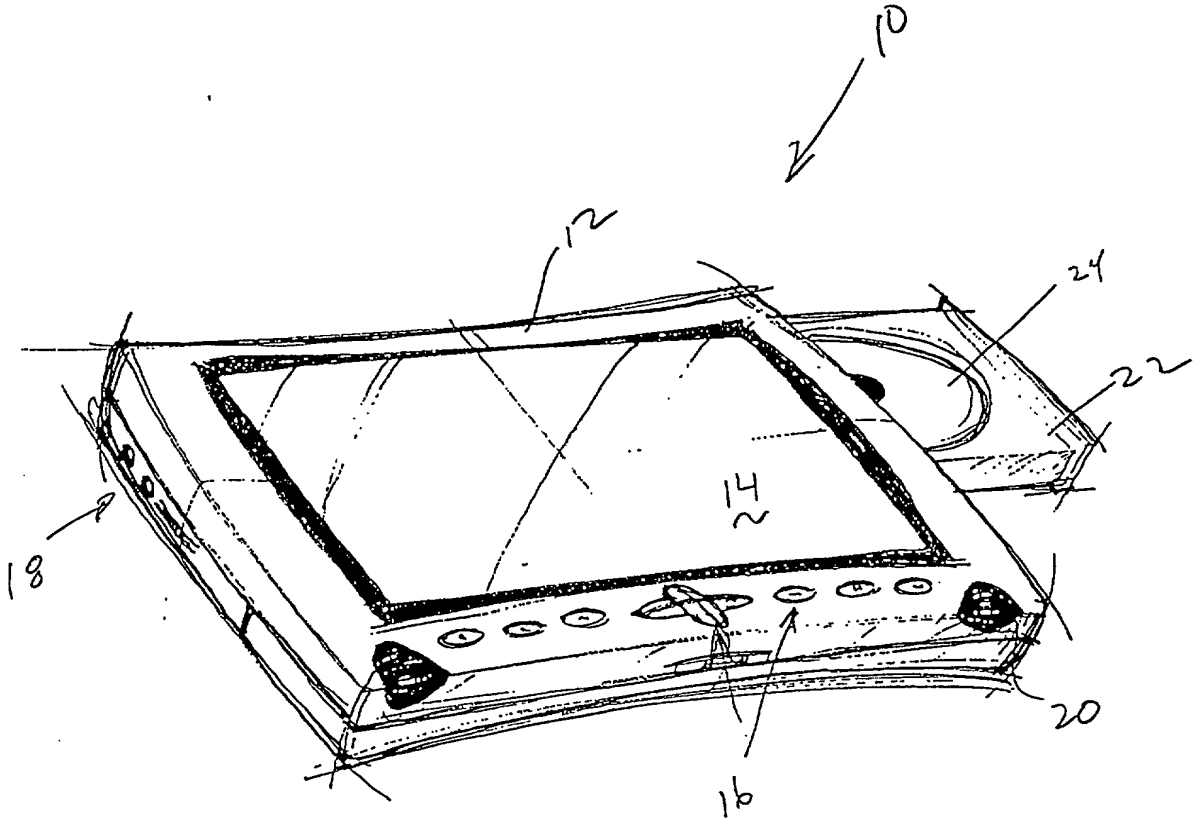


Fig. 1A

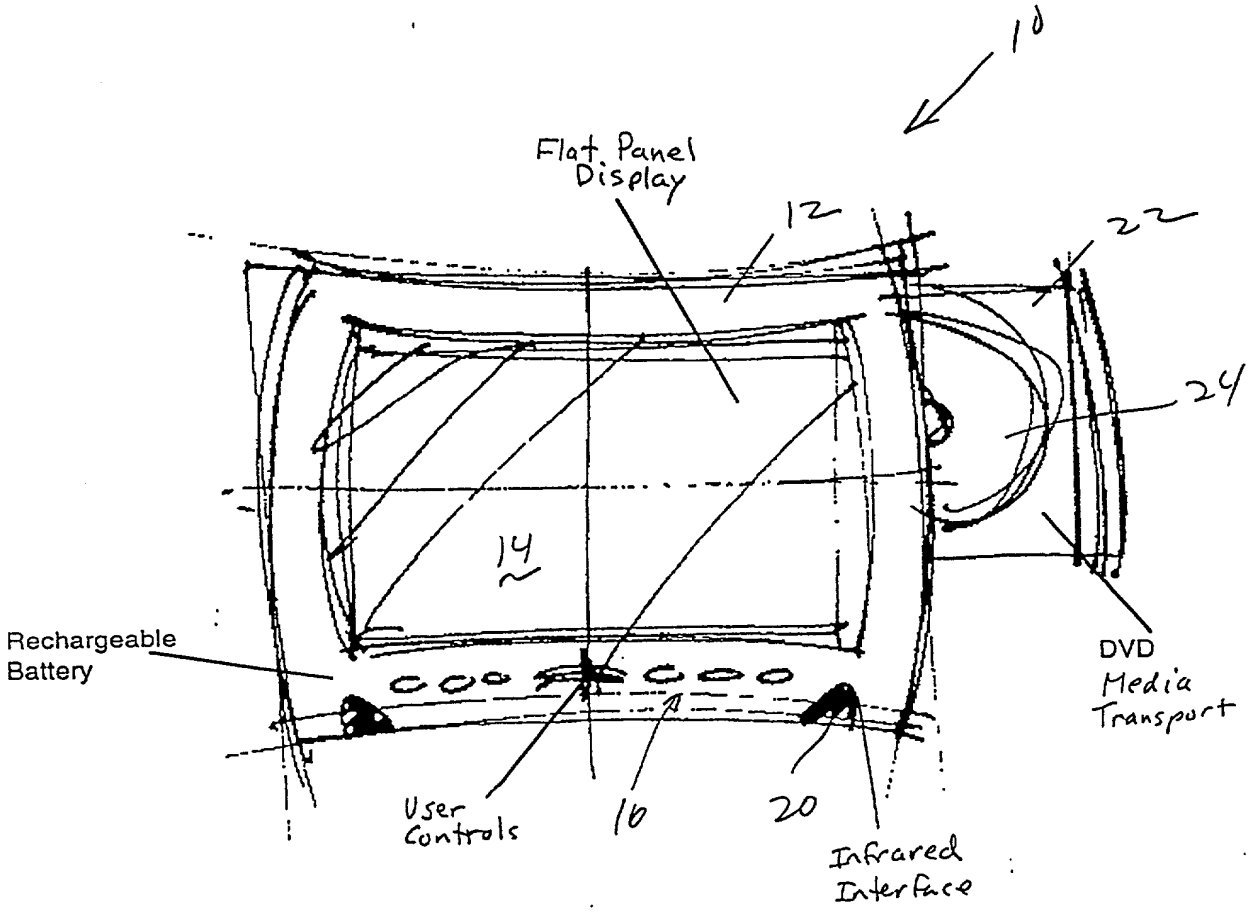

10
Flat Panel Display
12
22
24
14
Rechargeable Battery
DVD Media Transport
User Controls
16
20
Infrared Interface

Fig. 1B

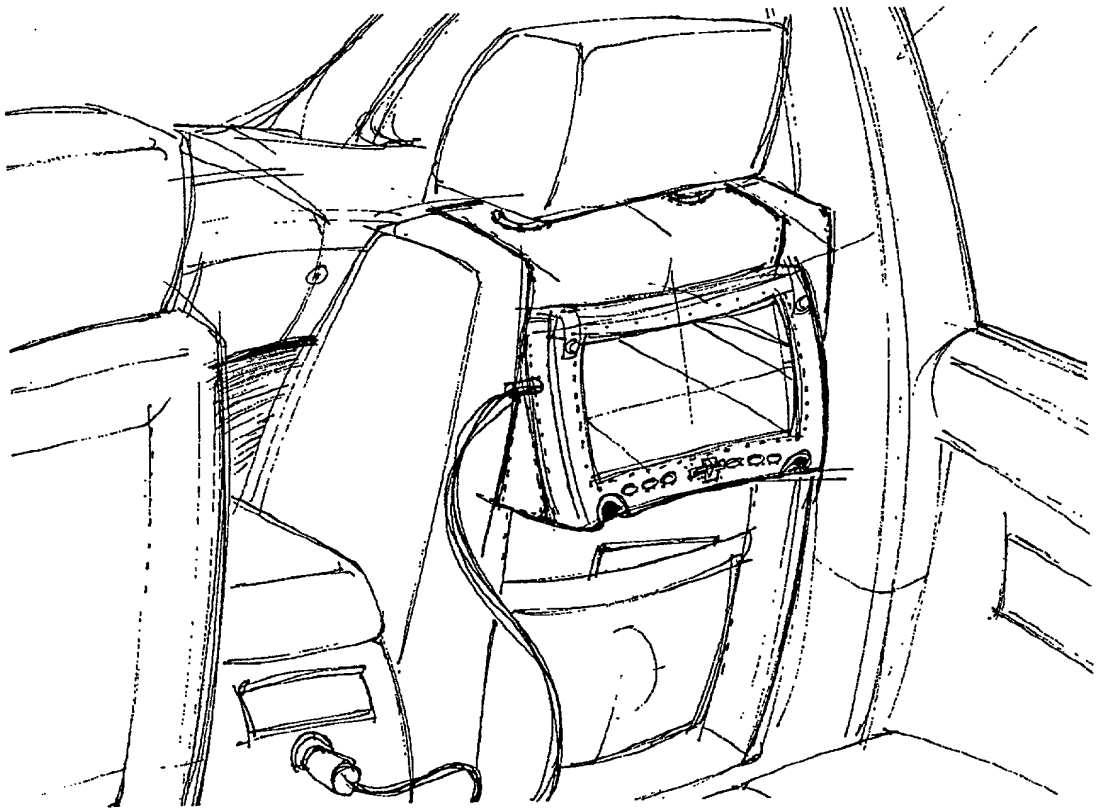


FIG. 2A

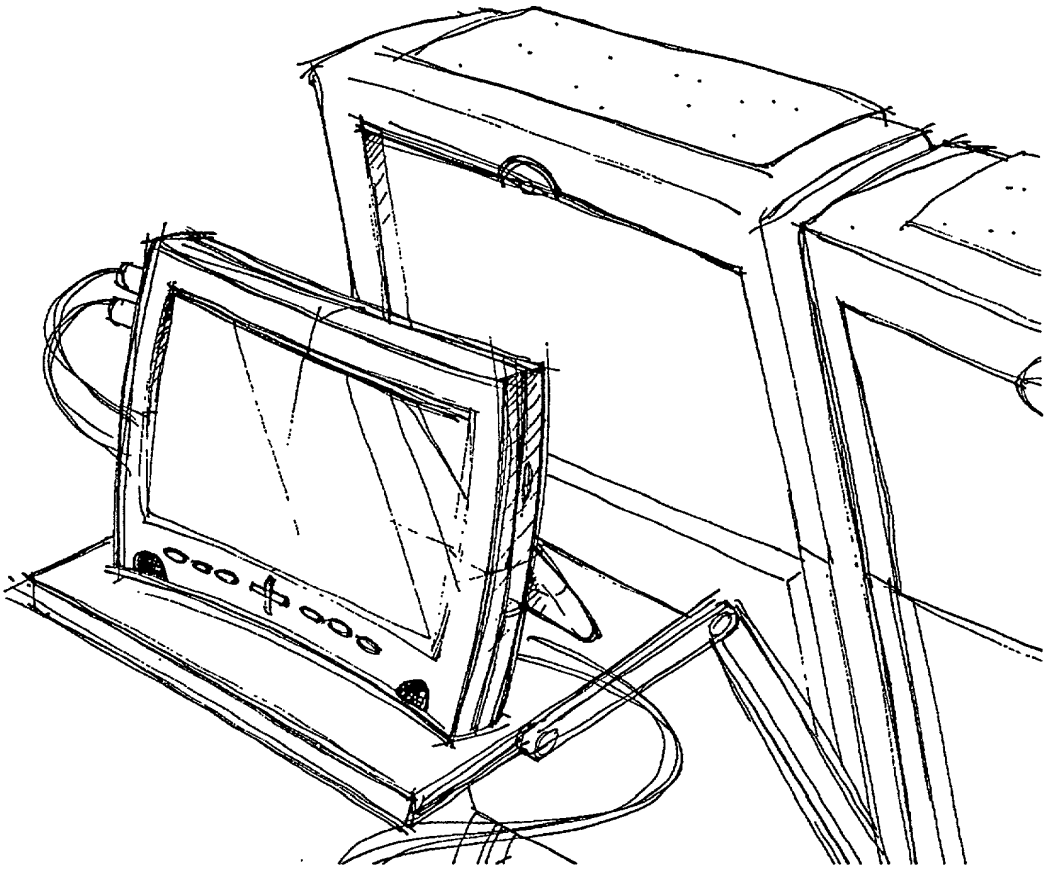


FIG. 2B

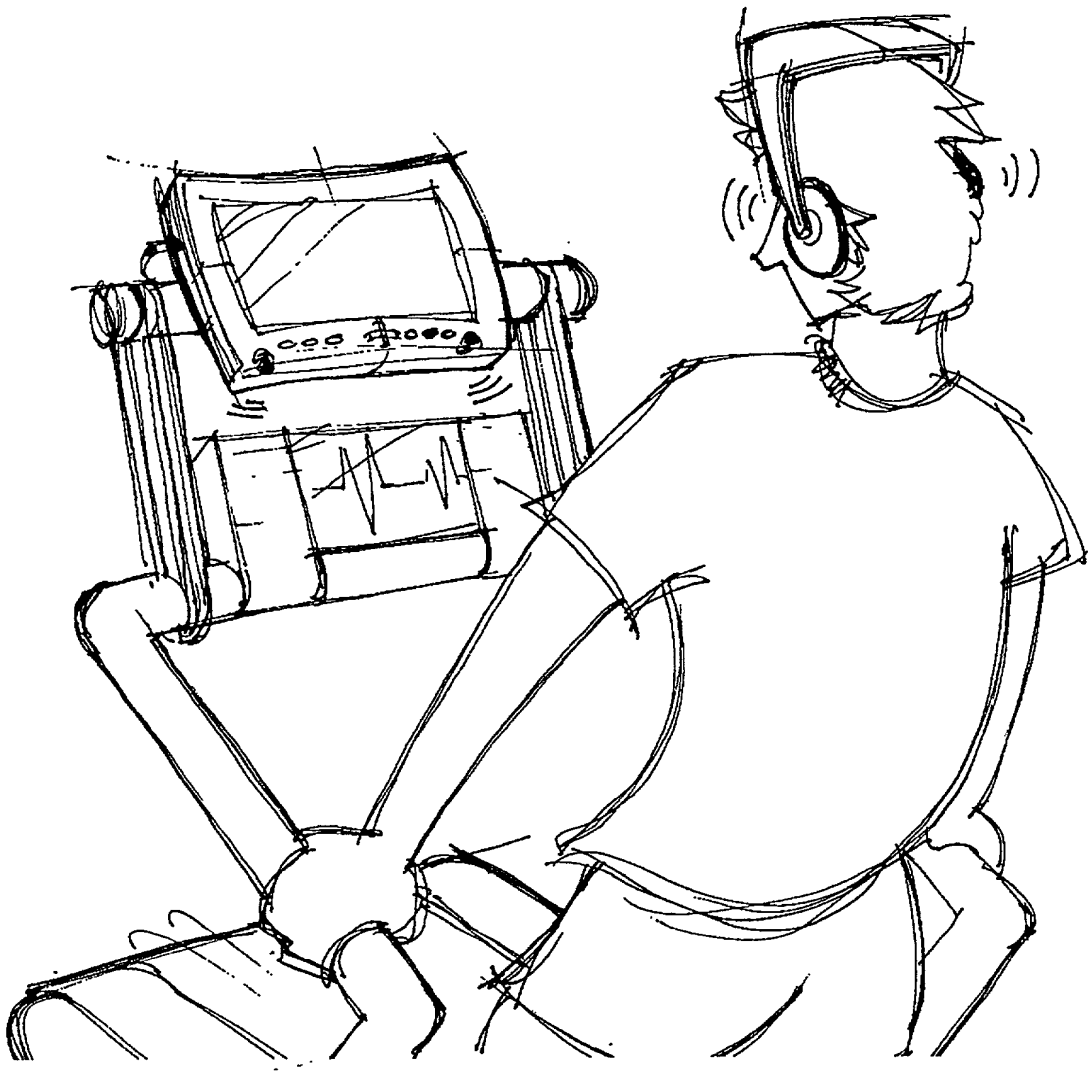


FIG. 2C

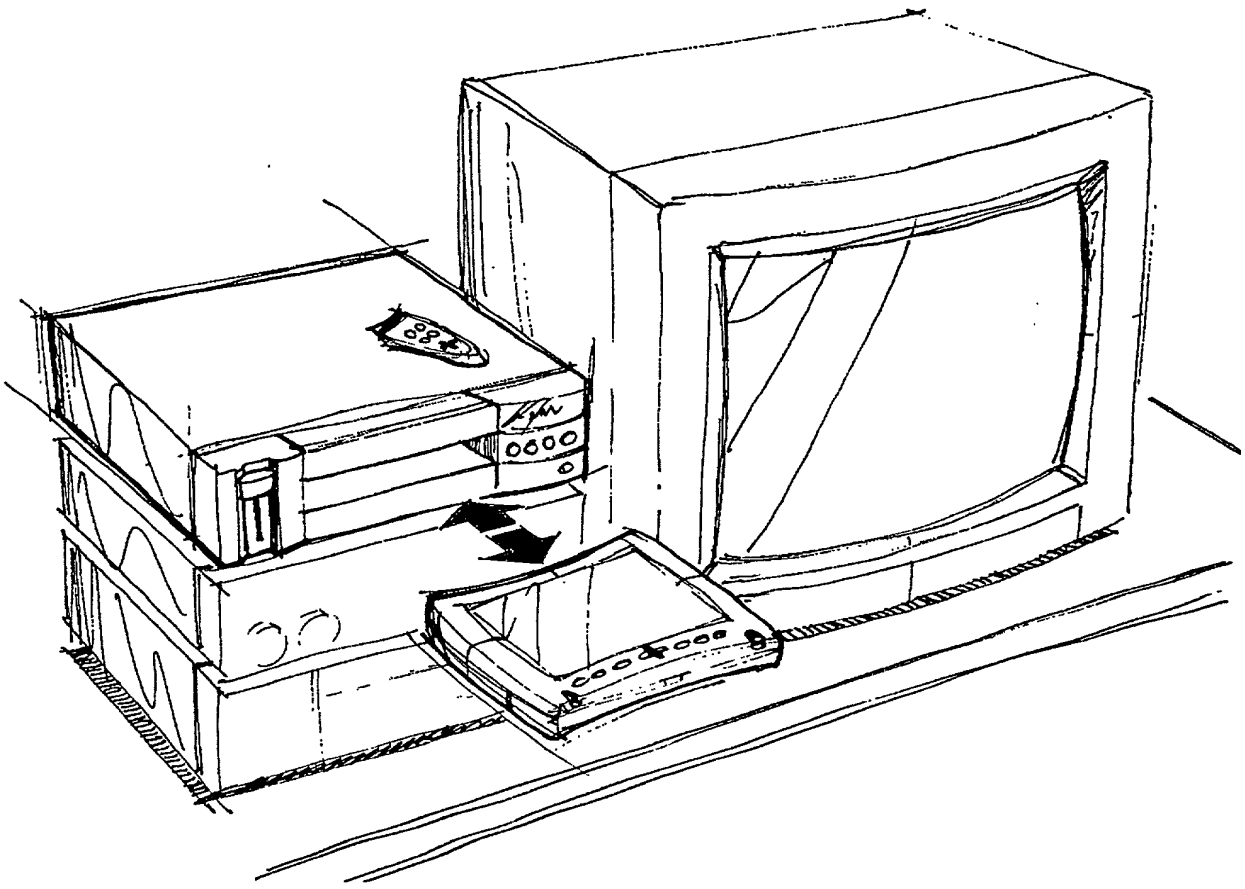


FIG. 2D

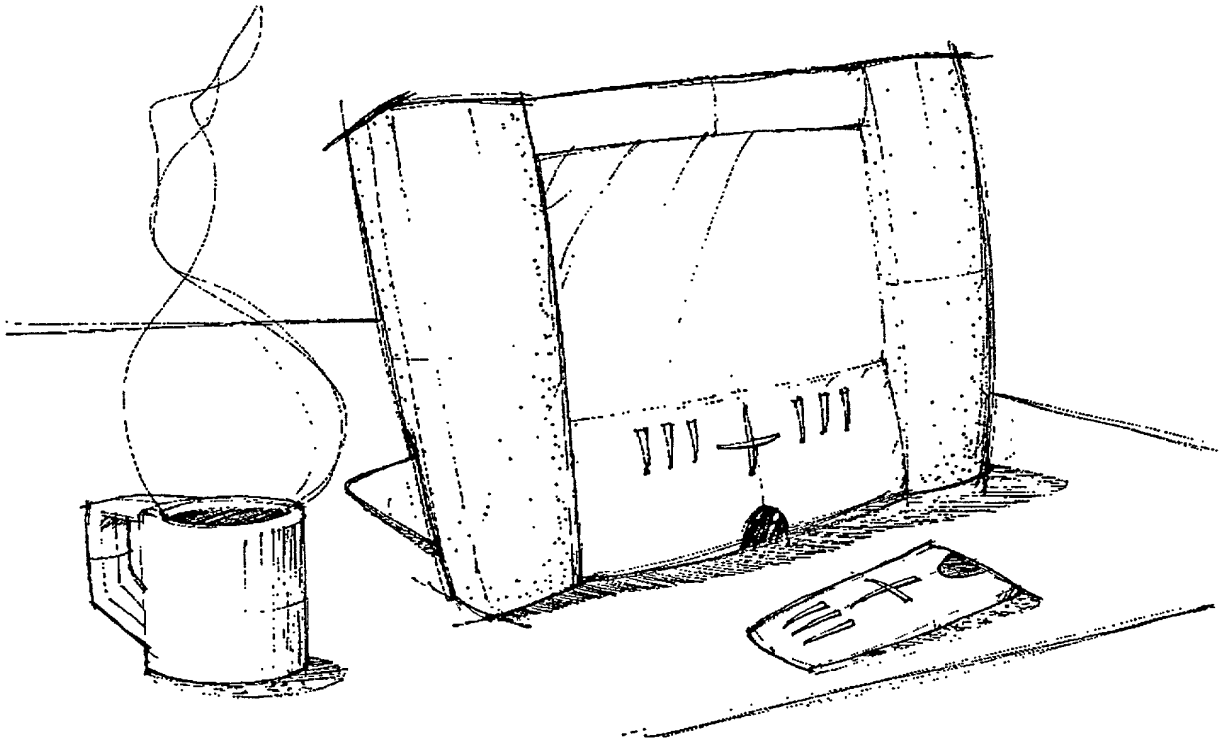


FIG. 2E

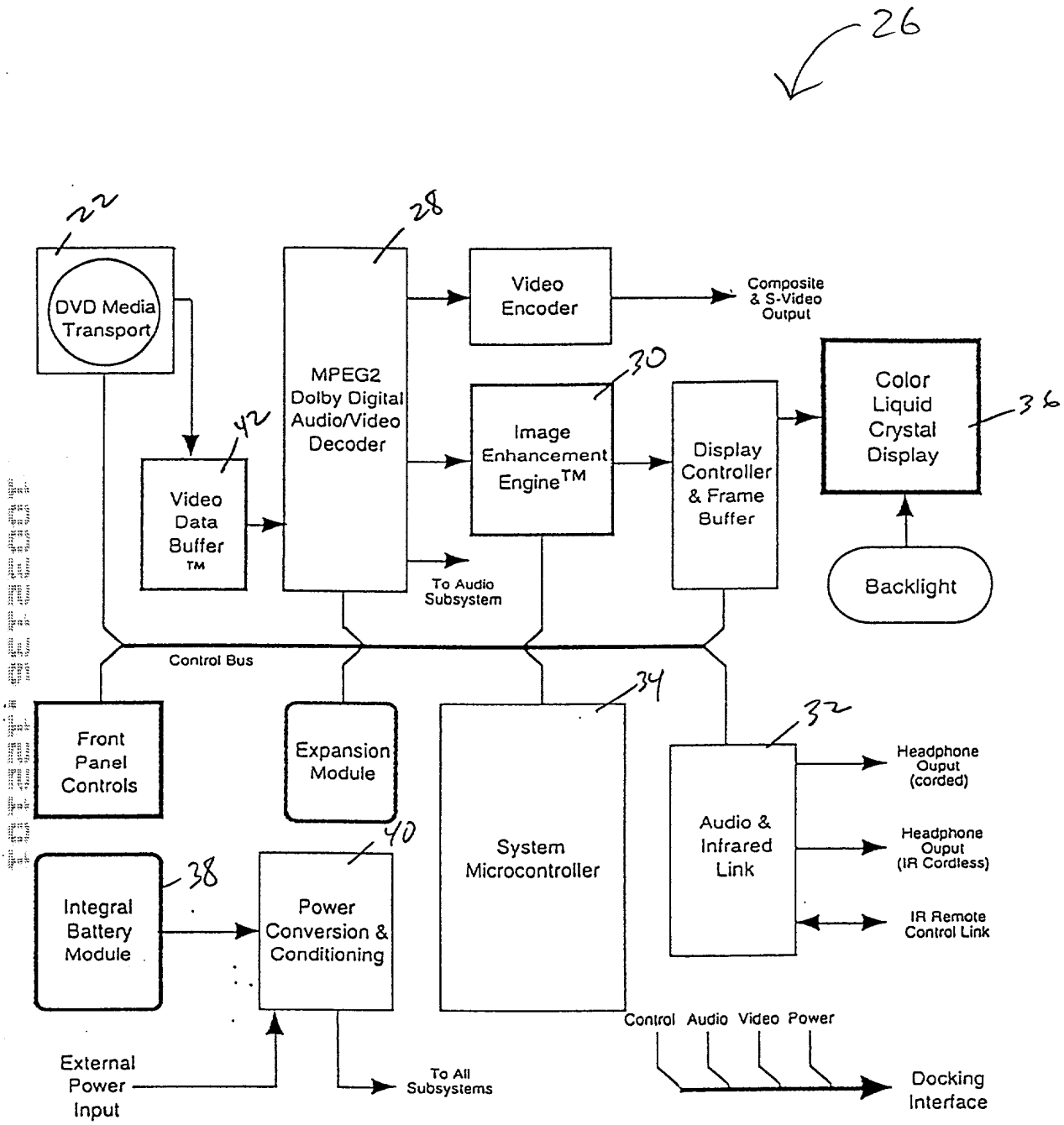

26
22
DVD Media Transport
28
MPEG2 Dolby Digital Audio/Video Decoder
Video Encoder
Composite & S-Video Output
30
Image Enhancement Engine™
Display Controller & Frame Buffer
Color Liquid Crystal Display
36
42
Video Data Buffer ™
To Audio Subsystem
Backlight
Control Bus
Front Panel Controls
Expansion Module
34
System Microcontroller
32
Audio & Infrared Link
Headphone Ouput (corded)
Headphone Ouput (IR Cordless)
IR Remote Control Link
38
Integral Battery Module
40
Power Conversion & Conditioning
External Power Input
To All Subsystems
Control Audio Video Power
Docking Interface

Fig. 3

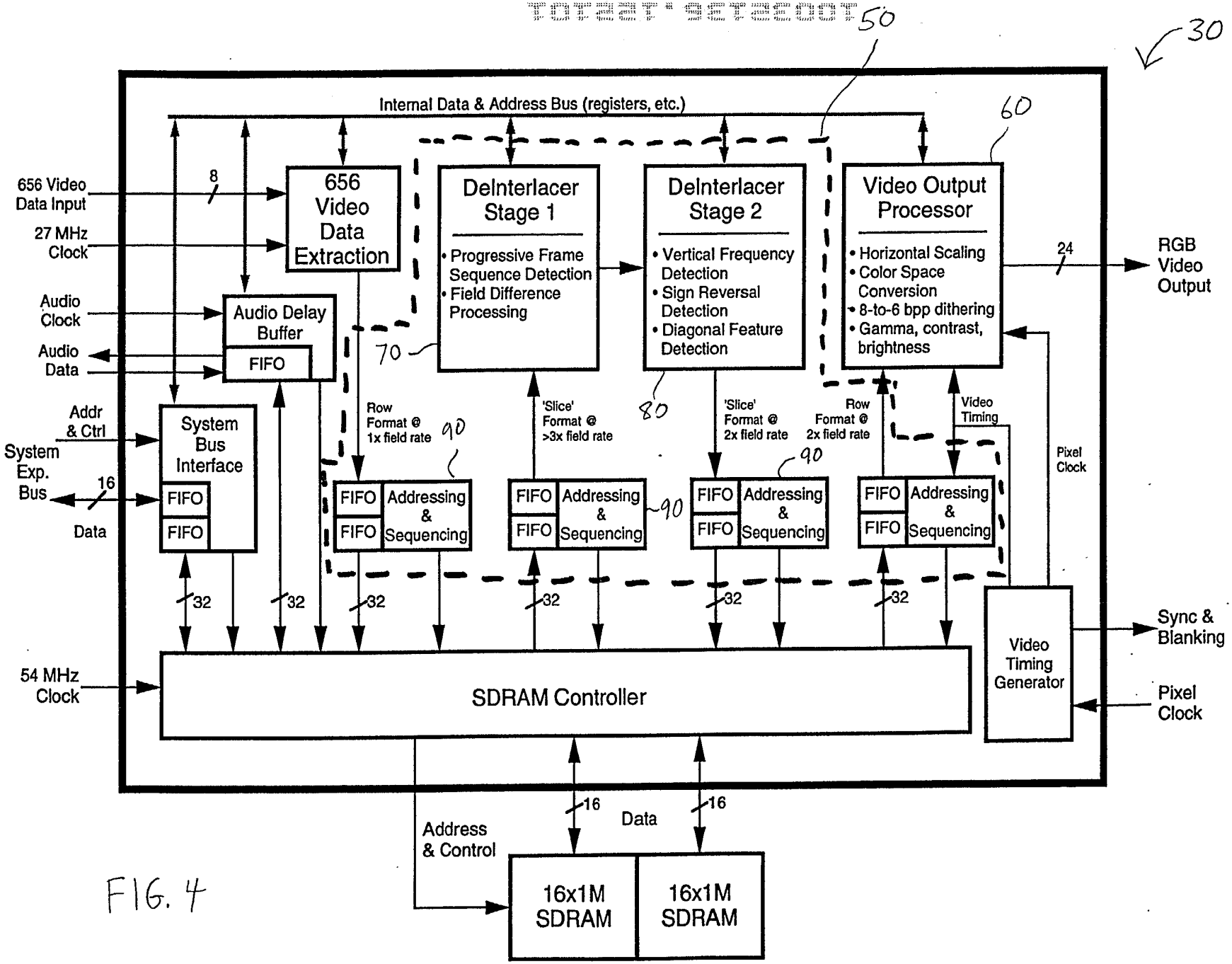


FIG. 4

100

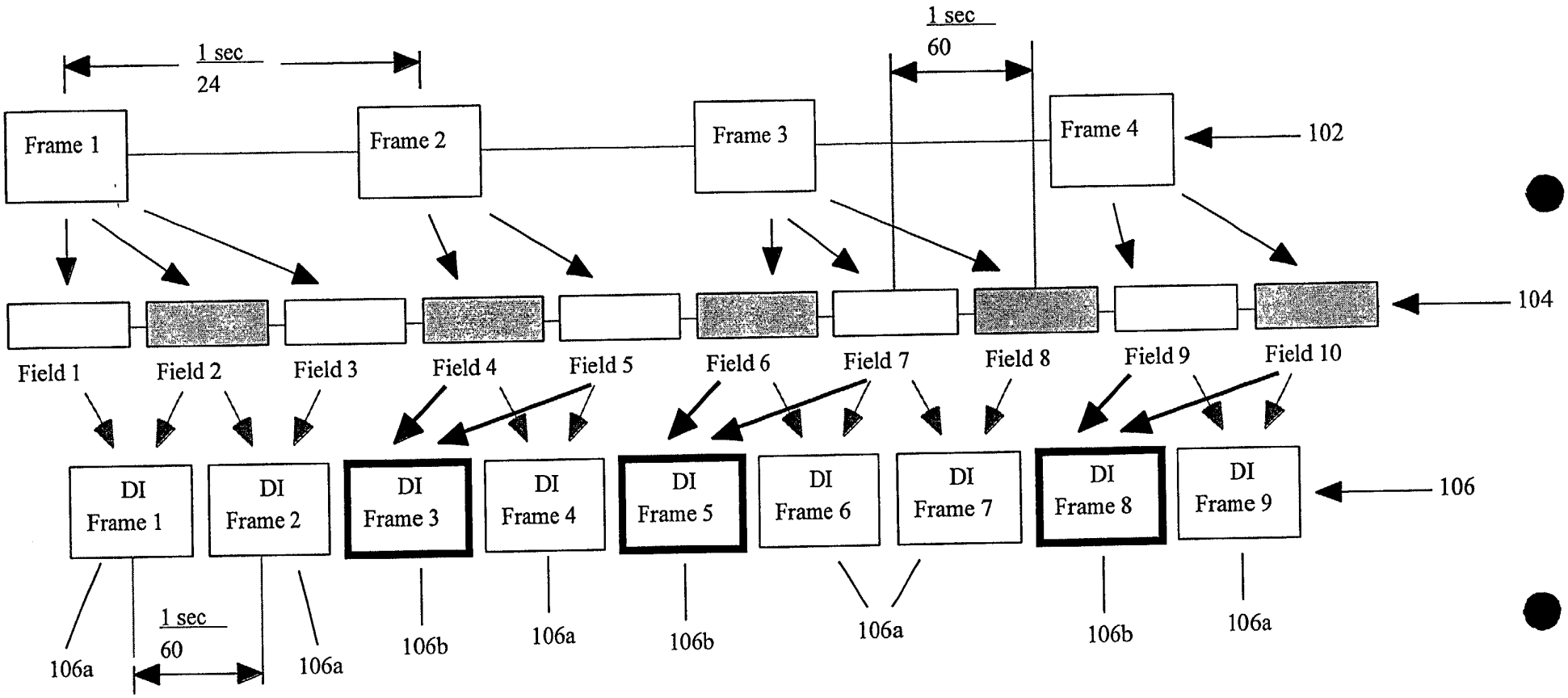

1 sec
24
1 sec
60
Frame 1
Frame 2
Frame 3
Frame 4
102
104
Field 1
Field 2
Field 3
Field 4
Field 5
Field 6
Field 7
Field 8
Field 9
Field 10
DI Frame 1
DI Frame 2
DI Frame 3
DI Frame 4
DI Frame 5
DI Frame 6
DI Frame 7
DI Frame 8
DI Frame 9
106
1 sec
60
106a
106a
106b
106a
106b
106a
106b
106a

Figure 5

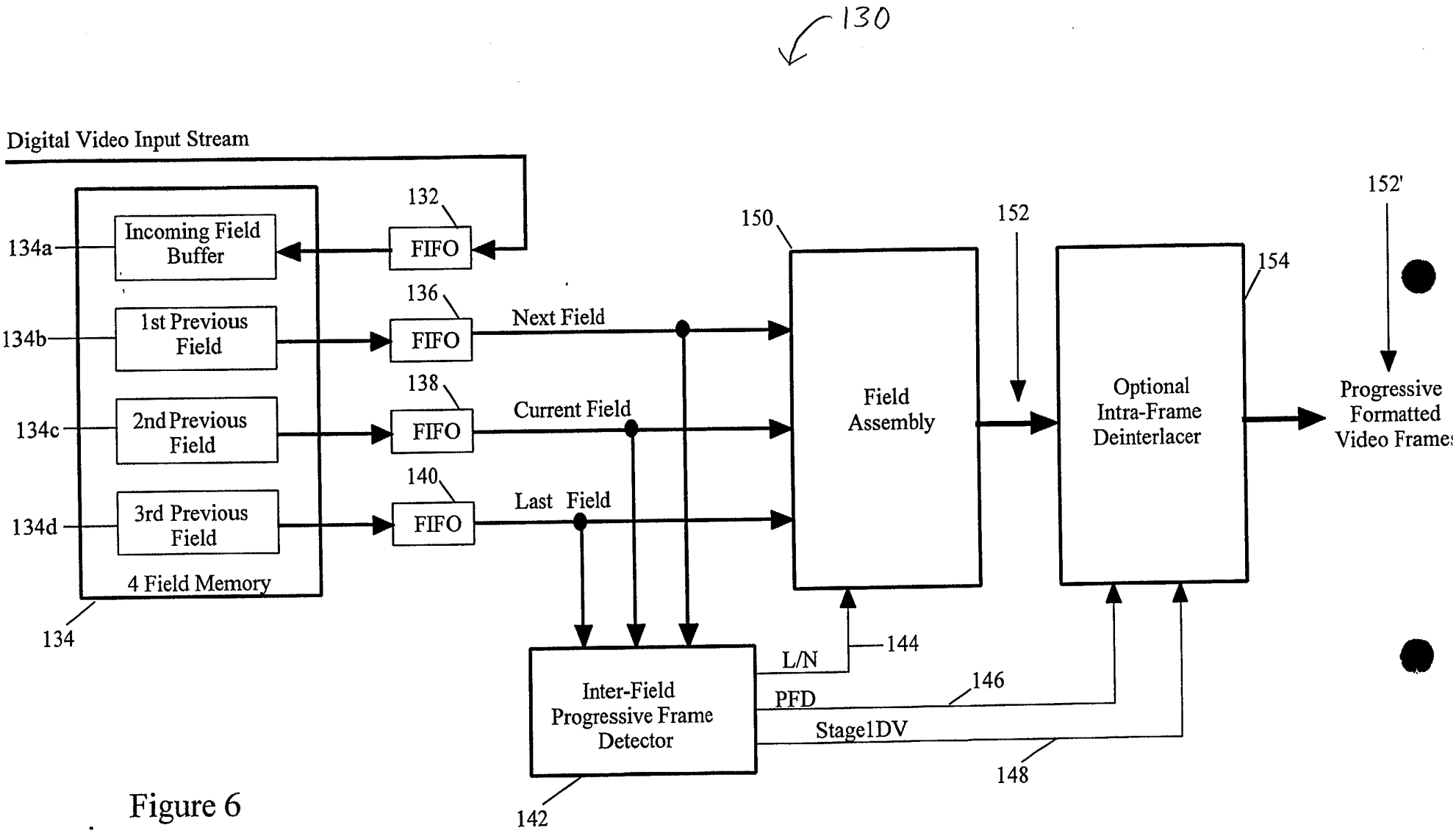

130
Digital Video Input Stream
132
FIFO
Incoming Field Buffer
134a
136
FIFO
1st Previous Field
134b
Next Field
138
FIFO
2nd Previous Field
134c
Current Field
140
FIFO
3rd Previous Field
134d
Last Field
4 Field Memory
134
150
Field Assembly
152
154
Optional Intra-Frame Deinterlacer
152'
Progressive Formatted Video Frames
Inter-Field Progressive Frame Detector
142
L/N
144
PFD
146
Stage1DV
148

Figure 6

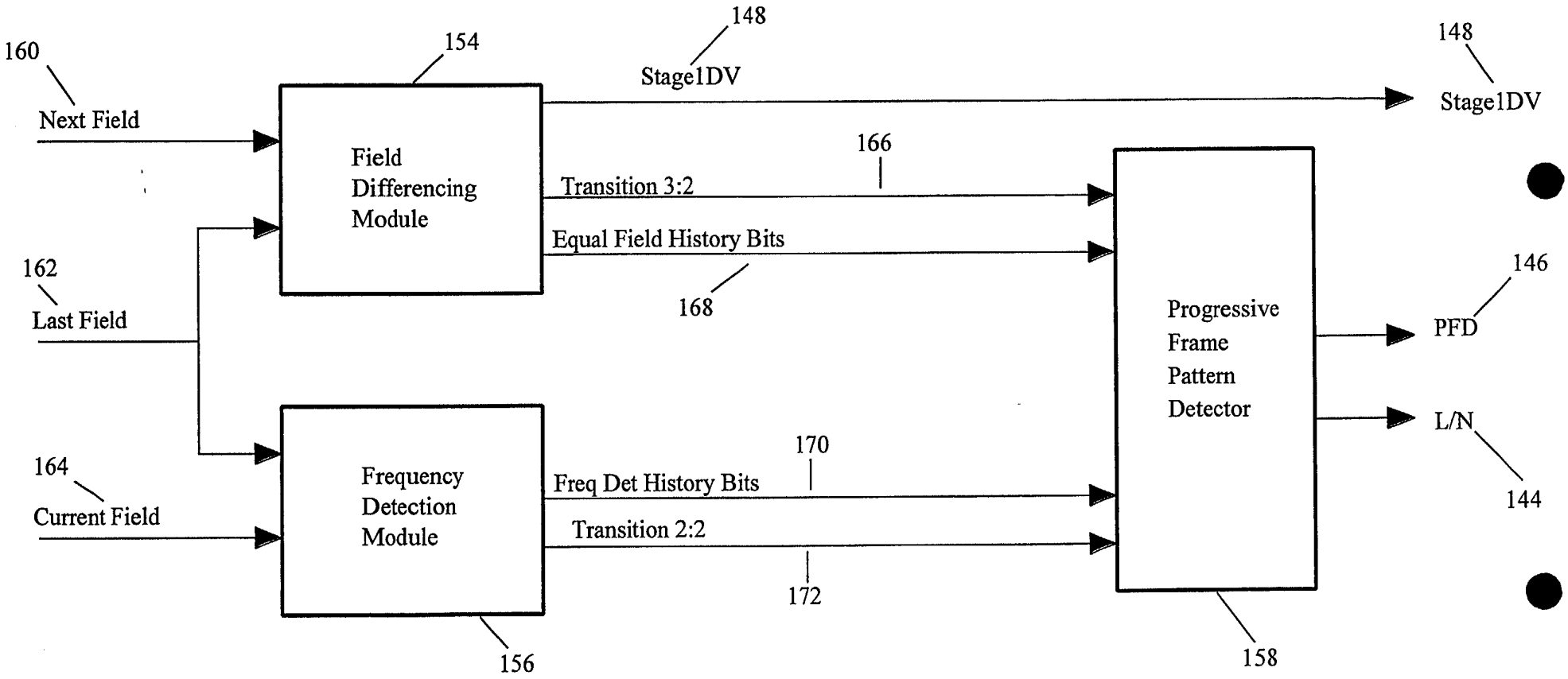

142
160
Next Field
154
Field
Differencing
Module
148
Stage1DV
148
Stage1DV
166
Transition 3:2
Equal Field History Bits
168
162
Last Field
164
Current Field
Frequency
Detection
Module
170
Freq Det History Bits
Transition 2:2
172
156
Progressive
Frame
Pattern
Detector
158
146
PFD
L/N
144

Figure 7

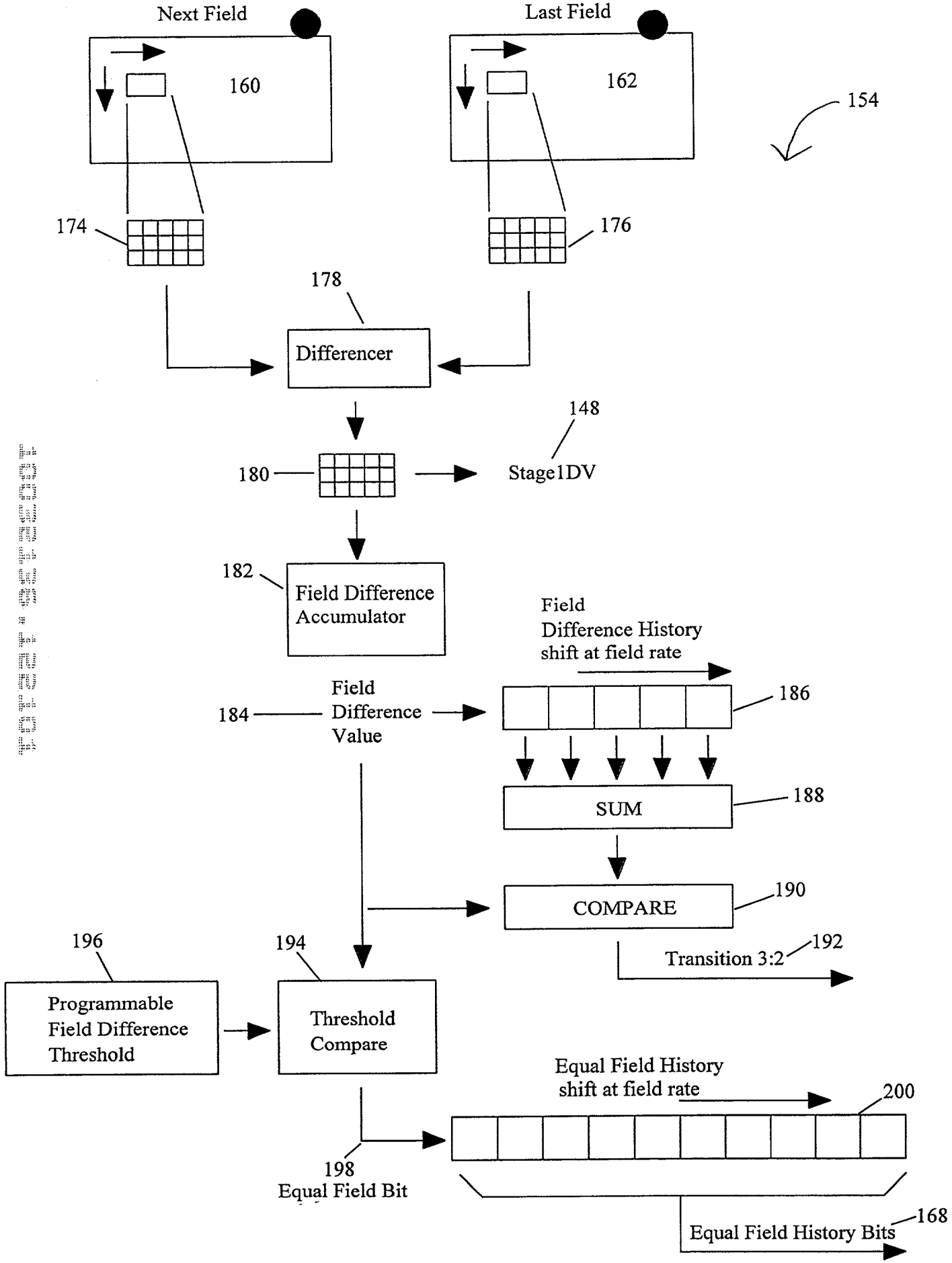

Next Field
160
Last Field
162
154
174
176
178
Differencer
148
180
Stage1DV
182
Field Difference Accumulator
Field
Difference History
shift at field rate
184
Field
Difference
Value
186
SUM
188
COMPARE
190
Transition 3:2
192
196
Programmable
Field Difference
Threshold
194
Threshold
Compare
Equal Field History
shift at field rate
200
198
Equal Field Bit
Equal Field History Bits
168

Figure 8

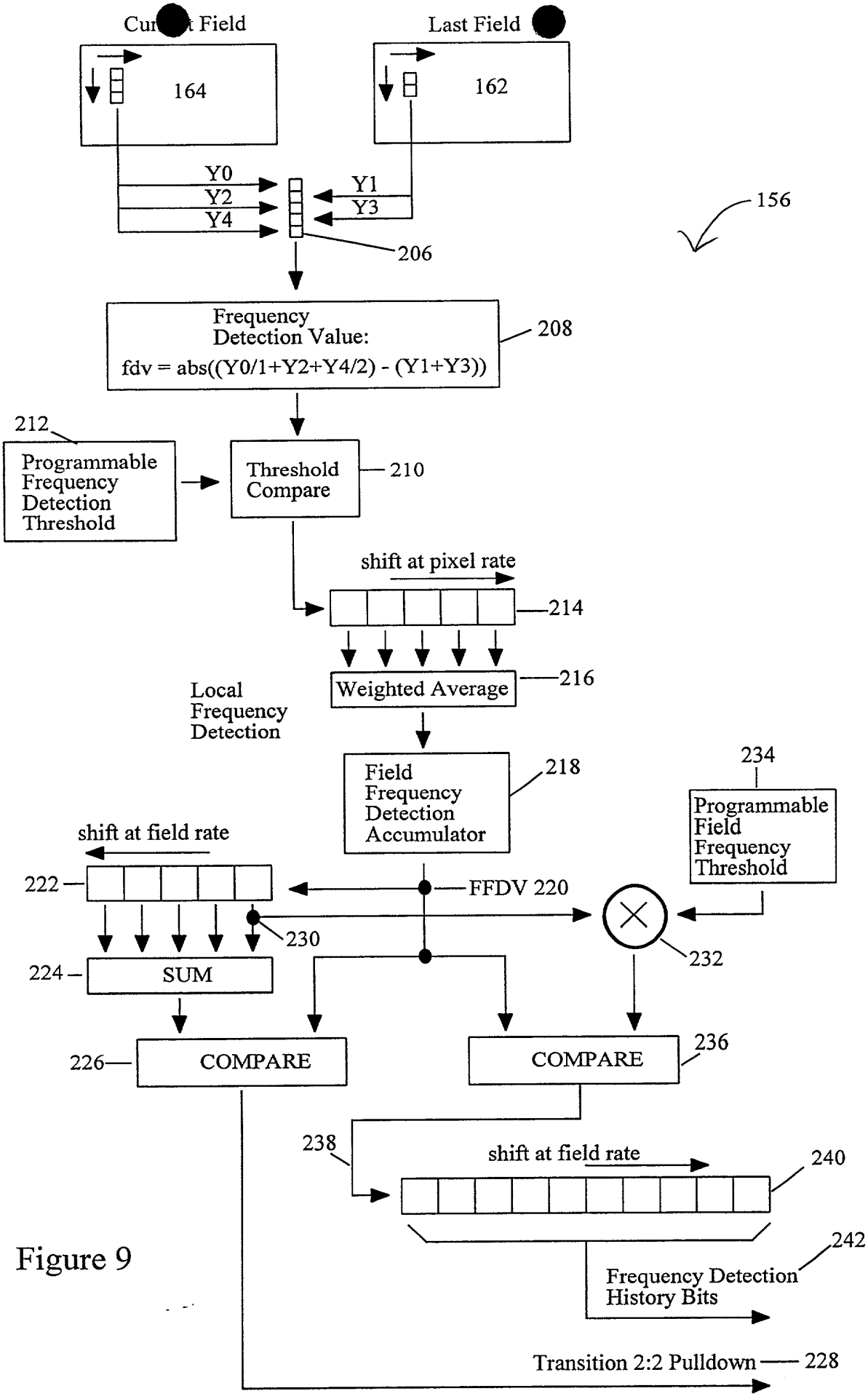

Current Field
Last Field
164
162
Y0
Y2
Y4
Y1
Y3
206
156
Frequency
Detection Value:
fdv = abs((Y0/1+Y2+Y4/2) - (Y1+Y3))
208
212
Programmable
Frequency
Detection
Threshold
Threshold
Compare
210
shift at pixel rate
214
Weighted Average
216
Local
Frequency
Detection
Field
Frequency
Detection
Accumulator
218
234
Programmable
Field
Frequency
Threshold
shift at field rate
222
FFDV 220
230
232
224
SUM
226
COMPARE
COMPARE
236
238
shift at field rate
240
242
Frequency Detection
History Bits
Transition 2:2 Pulldown
228

Figure 9

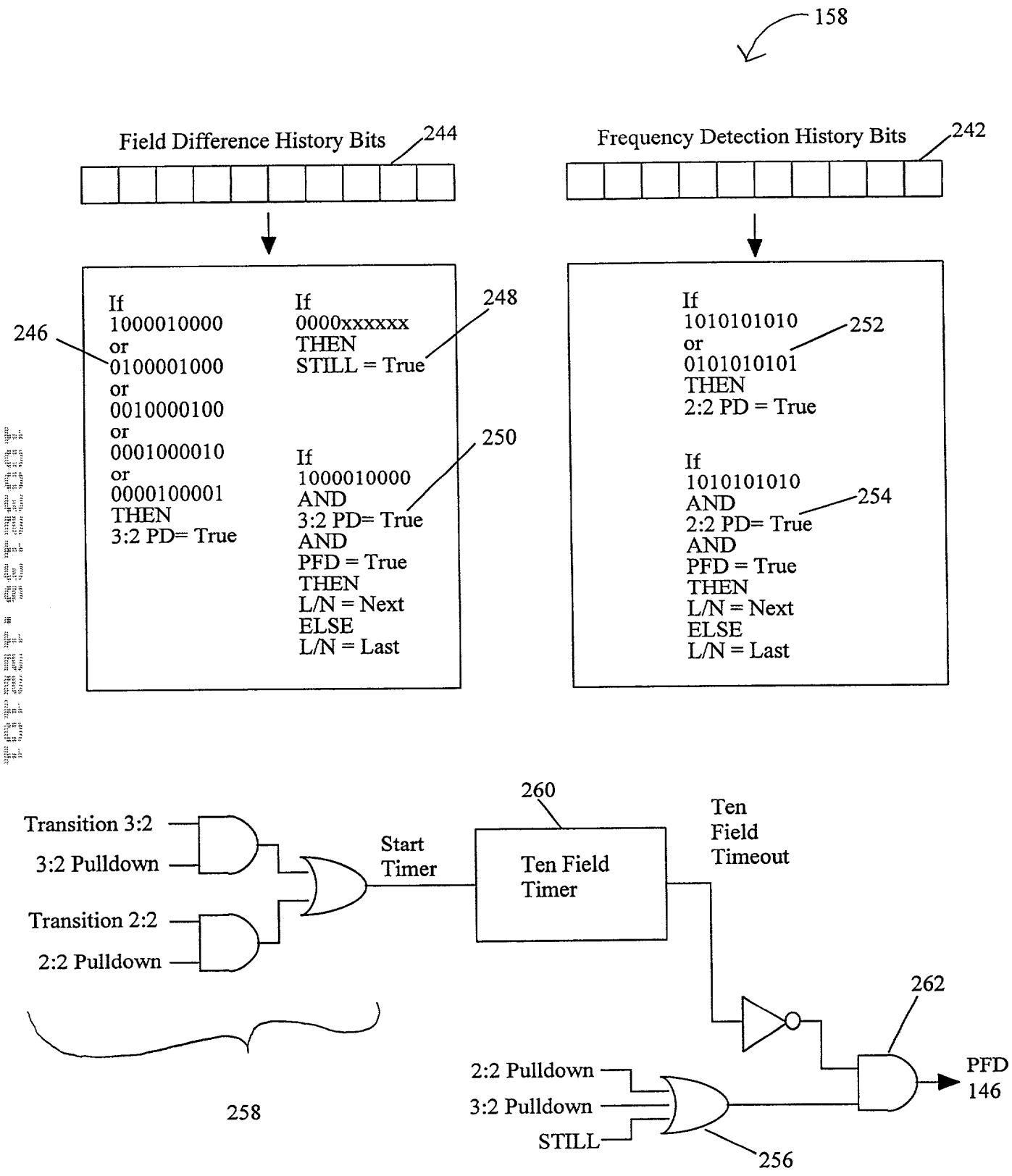

158
Field Difference History Bits
244
Frequency Detection History Bits
242
If
1000010000
or
0100001000
or
0010000100
or
0001000010
or
0000100001
THEN
3:2 PD= True
246
If
0000xxxxxx
THEN
STILL = True
248
If
1000010000
AND
3:2 PD= True
AND
PFD = True
THEN
L/N = Next
ELSE
L/N = Last
250
If
1010101010
or
0101010101
THEN
2:2 PD = True
252
If
1010101010
AND
2:2 PD= True
AND
PFD = True
THEN
L/N = Next
ELSE
L/N = Last
254
Transition 3:2
3:2 Pulldown
Transition 2:2
2:2 Pulldown
Start
Timer
260
Ten Field
Timer
Ten
Field
Timeout
262
PFD
146
2:2 Pulldown
3:2 Pulldown
STILL
256
258

Figure 10

Figure 11

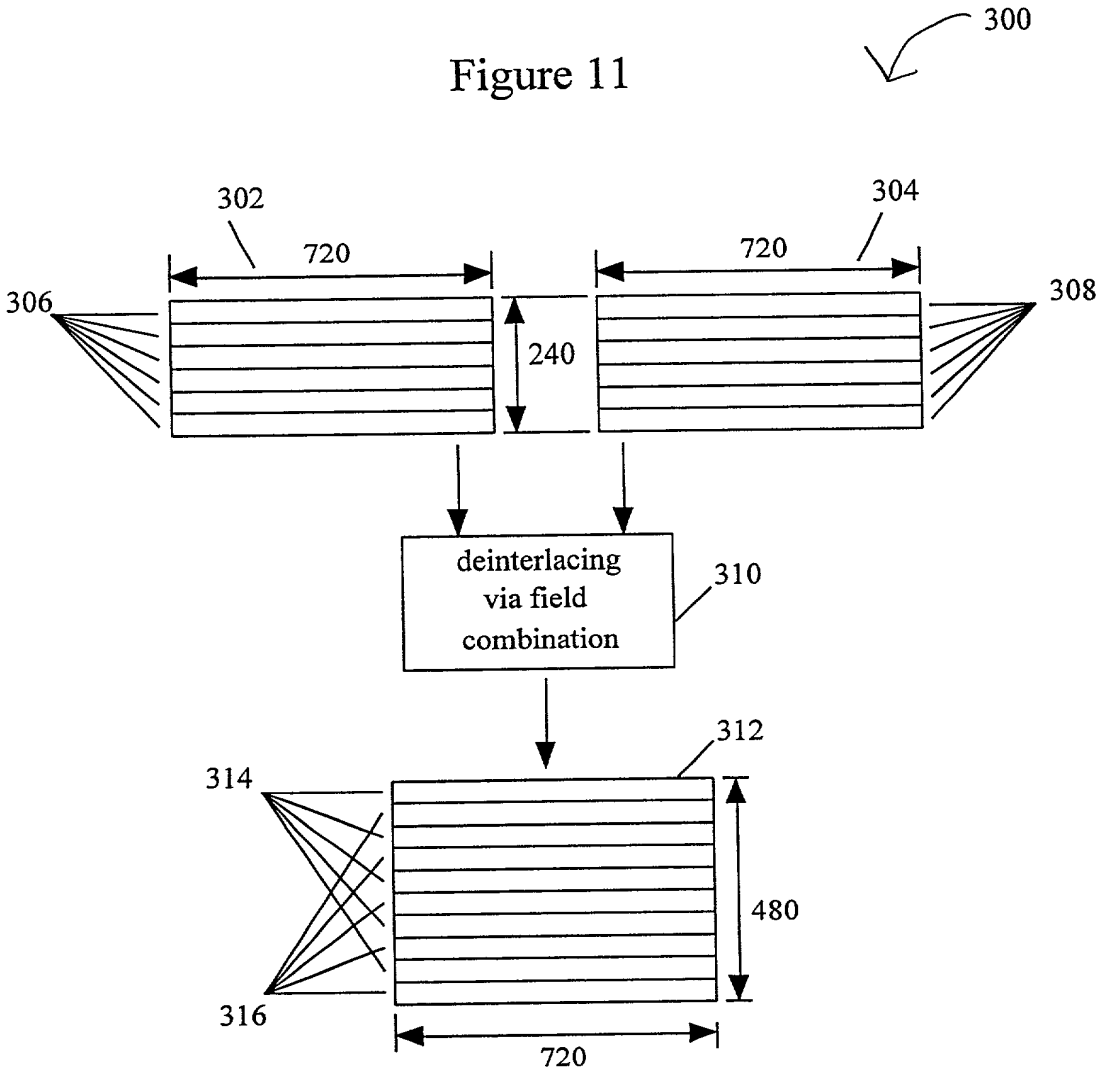
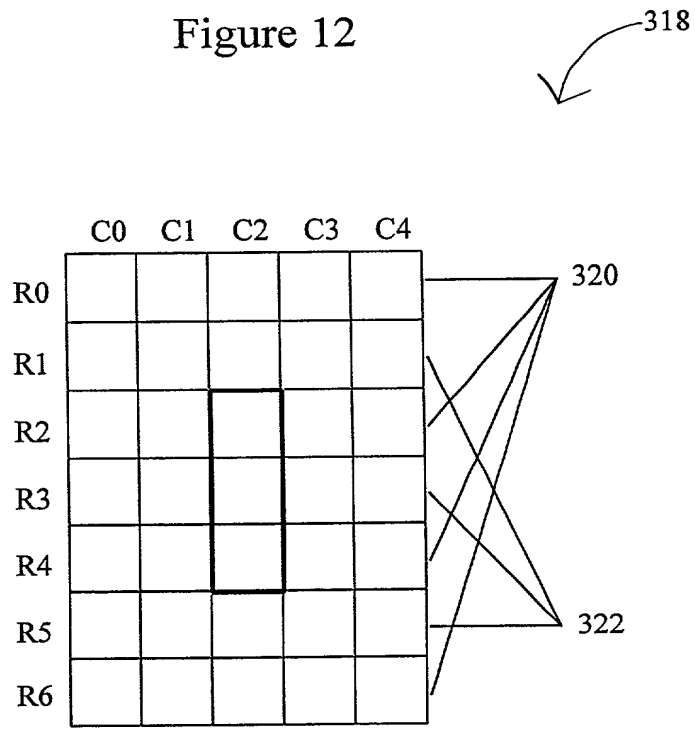

300
302
304
720
720
306
308
240
deinterlacing
via field
combination
310
312
314
480
316
720

Figure 12



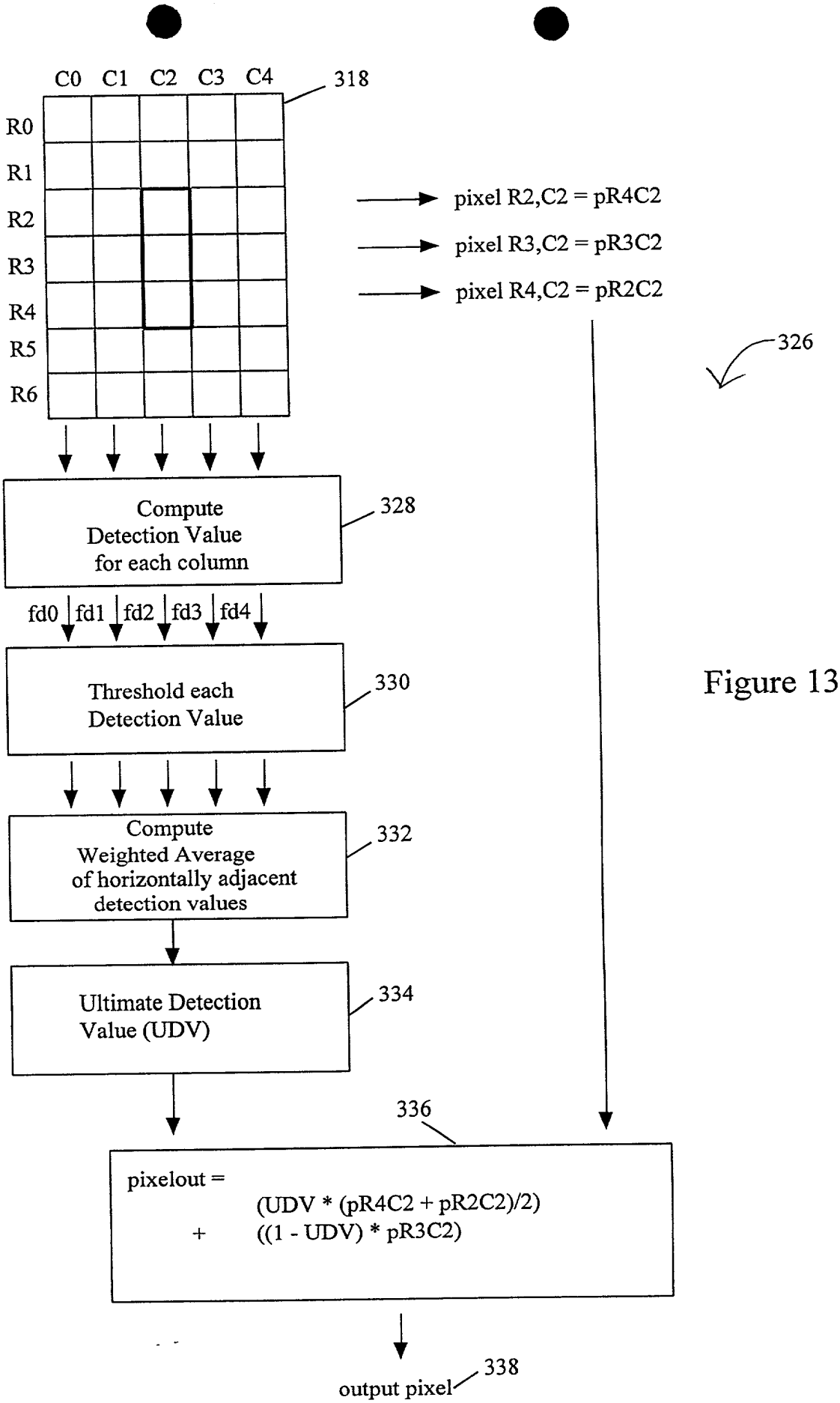

C0 C1 C2 C3 C4
318
R0
R1
R2
R3
R4
R5
R6
pixel R2,C2 = pR4C2
pixel R3,C2 = pR3C2
pixel R4,C2 = pR2C2
326
Compute Detection Value for each column
328
fd0 fd1 fd2 fd3 fd4
Threshold each Detection Value
330
Compute Weighted Average of horizontally adjacent detection values
332
Ultimate Detection Value (UDV)
334
336
pixelout =
(UDV * (pR4C2 + pR2C2)/2)
+ ((1 - UDV) * pR3C2)
output pixel
338

Figure 13

Figure 14A

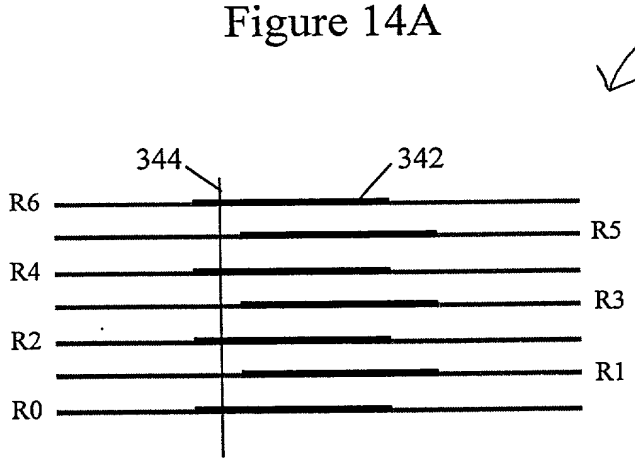


Figure 14B

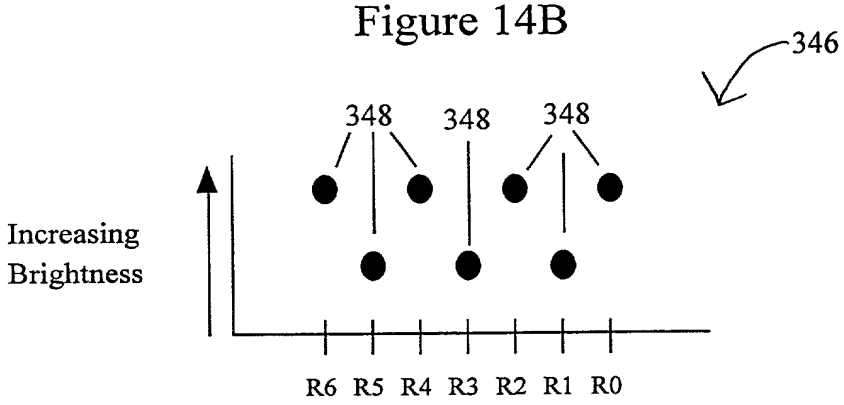


Figure 14C

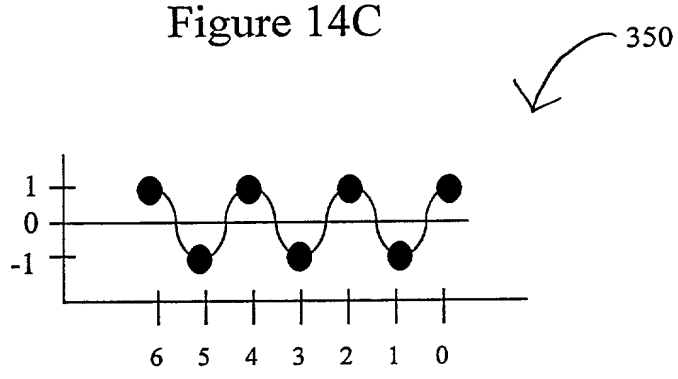


Figure 15

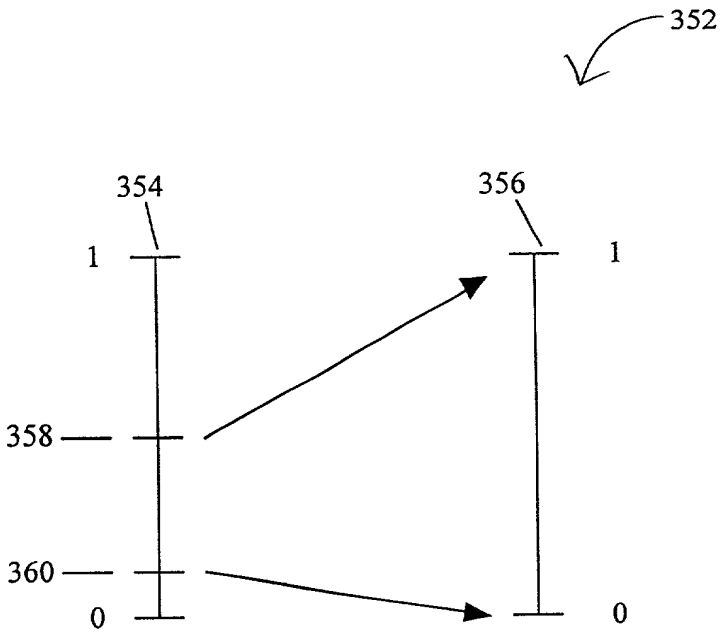


Figure 16

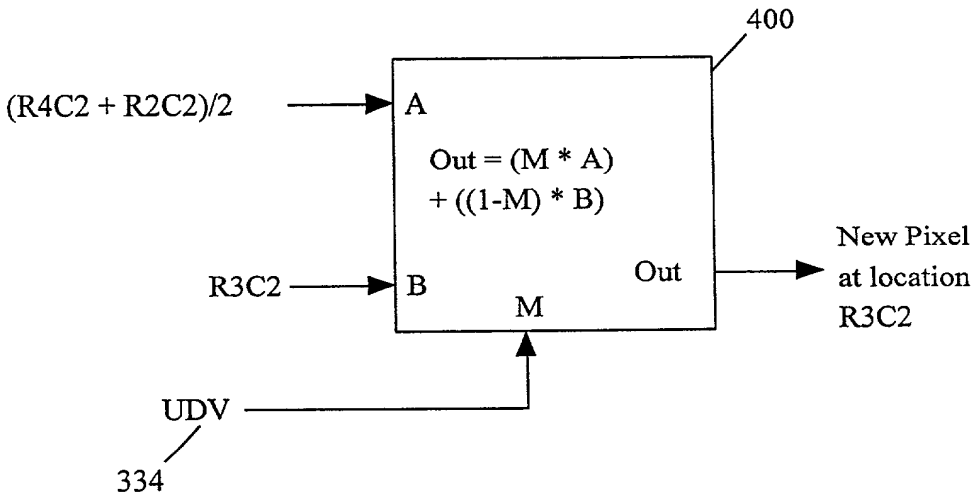


Figure 17

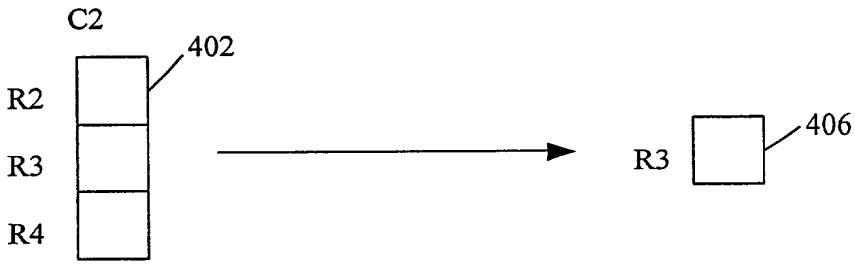


Figure 18

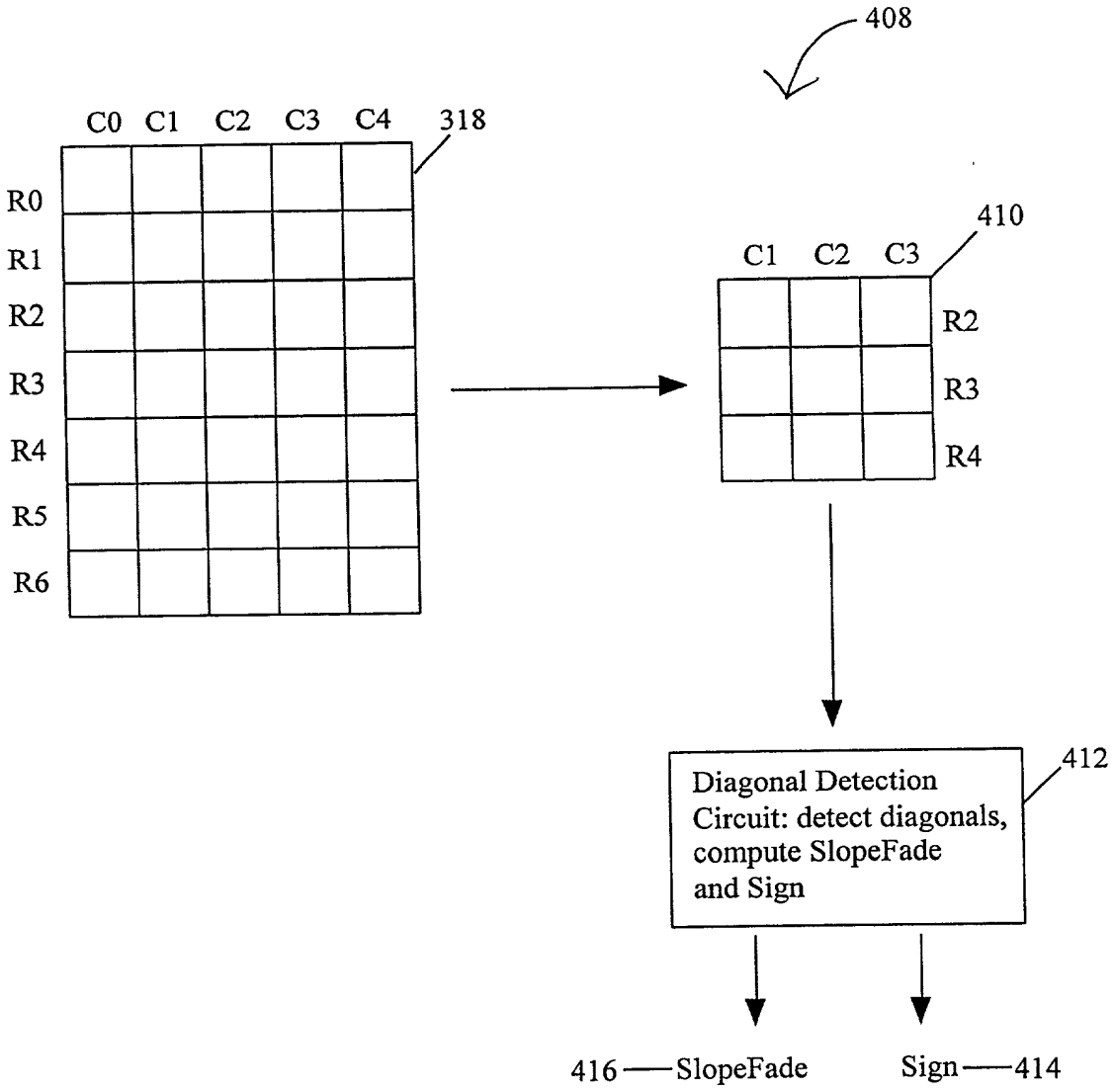

408
C0 C1 C2 C3 C4
318
R0
R1
R2
R3
R4
R5
R6
410
C1 C2 C3
R2
R3
R4
Diagonal Detection Circuit: detect diagonals, compute SlopeFade and Sign
412
416 SlopeFade
Sign 414

Figure 19

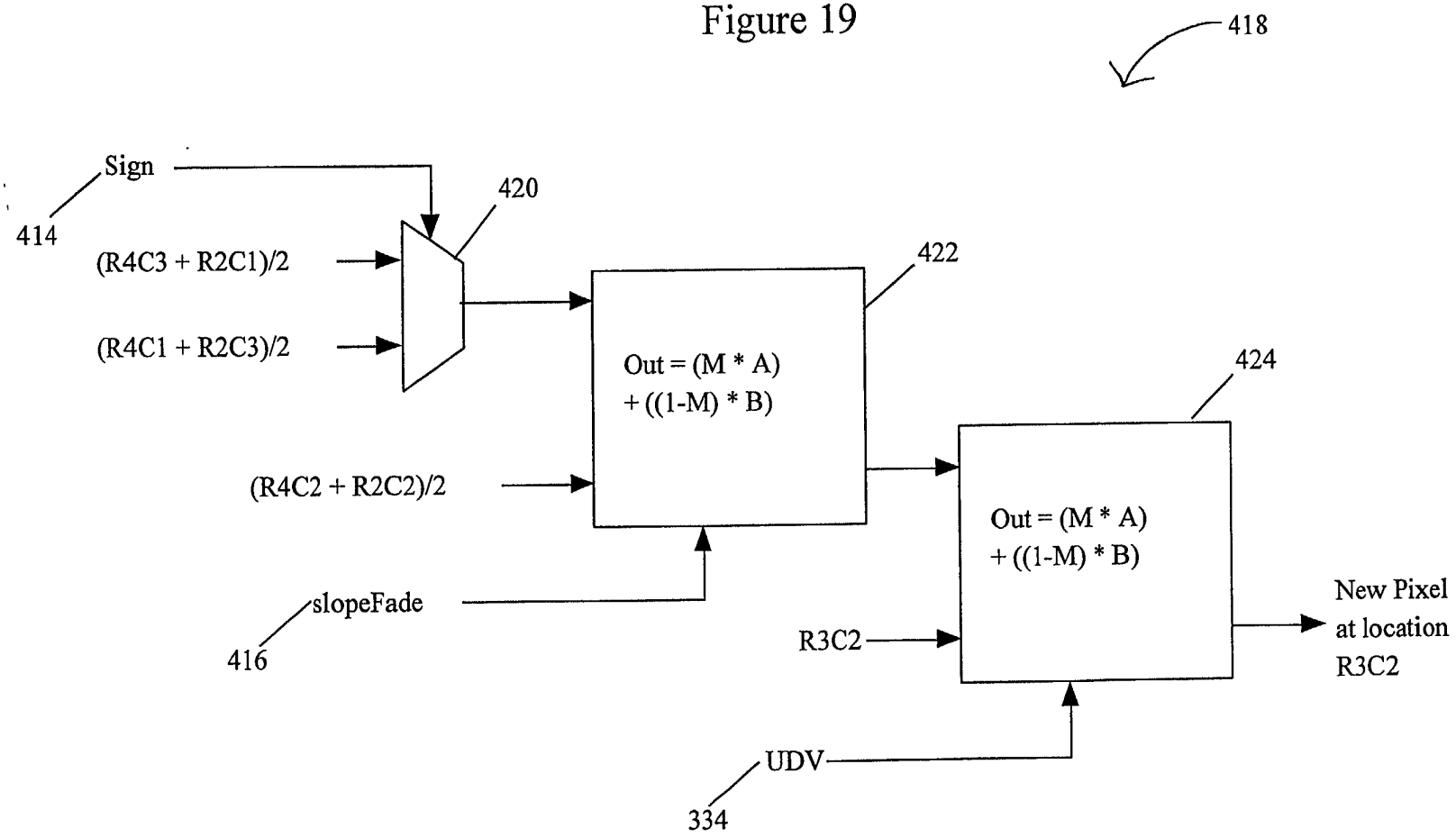

418
Sign
414
420
(R4C3 + R2C1)/2
(R4C1 + R2C3)/2
422
Out = (M * A)
+ ((1-M) * B)
424
(R4C2 + R2C2)/2
Out = (M * A)
+ ((1-M) * B)
New Pixel
at location
R3C2
slopeFade
416
R3C2
UDV
334

Figure 20

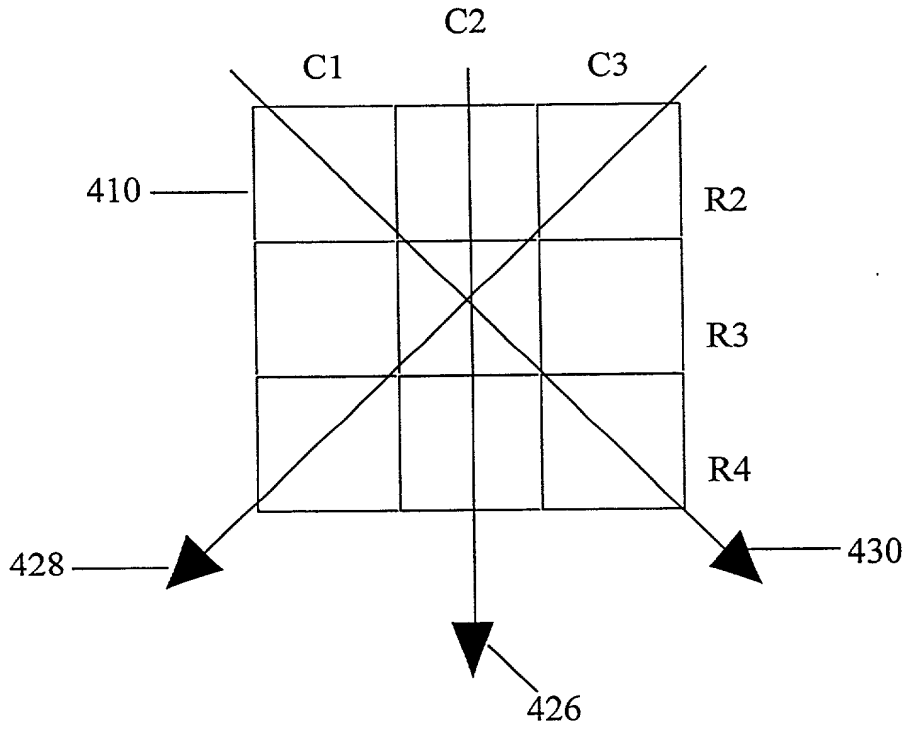
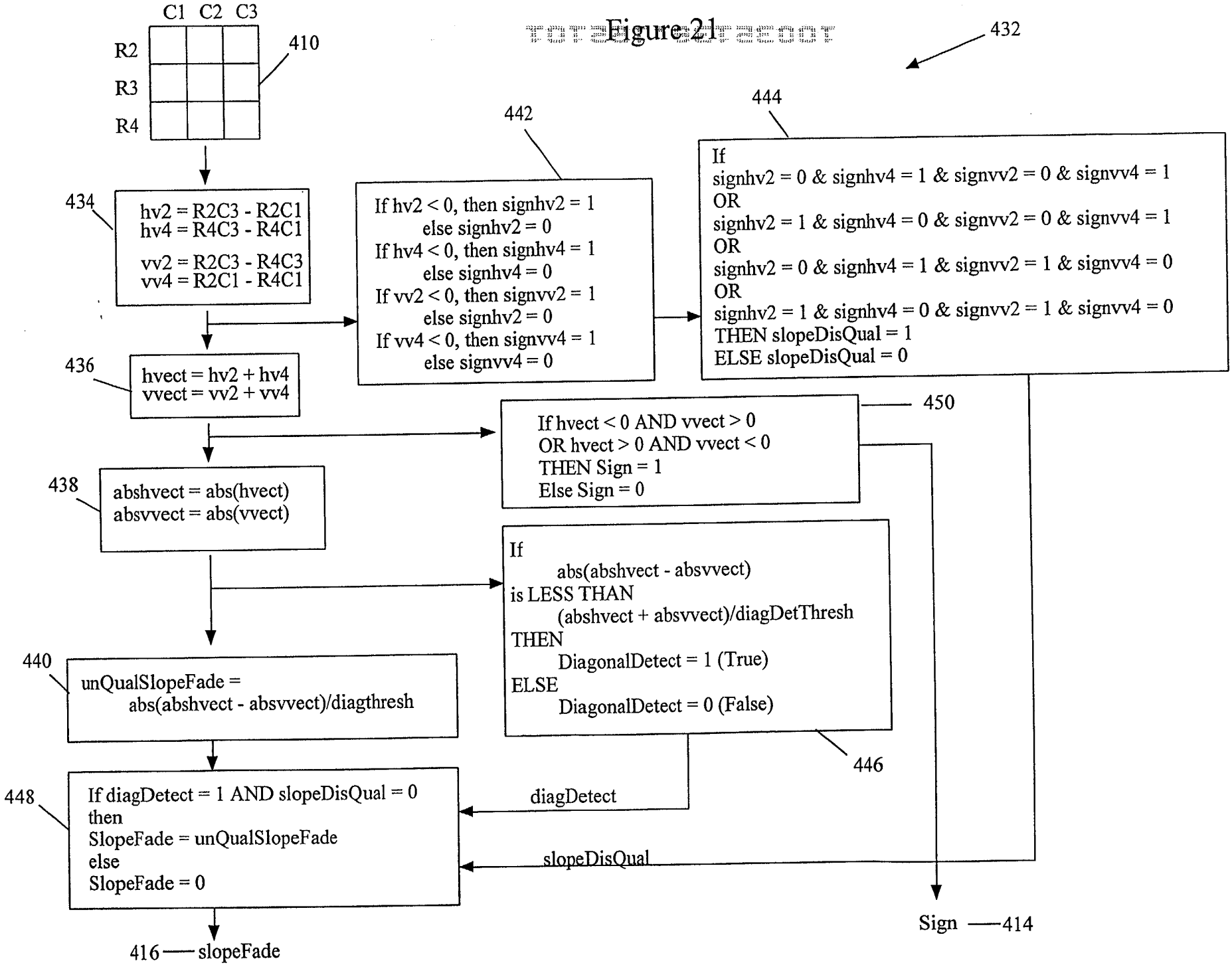


Figure 21


C1 C2 C3
R2
R3
R4
410
432
434
hv2 = R2C3 - R2C1
hv4 = R4C3 - R4C1
vv2 = R2C3 - R4C3
vv4 = R2C1 - R4C1
442
If hv2 < 0, then signhv2 = 1
else signhv2 = 0
If hv4 < 0, then signhv4 = 1
else signhv4 = 0
If vv2 < 0, then signvv2 = 1
else signhv2 = 0
If vv4 < 0, then signvv4 = 1
else signvv4 = 0
444
If
signhv2 = 0 & signhv4 = 1 & signvv2 = 0 & signvv4 = 1
OR
signhv2 = 1 & signhv4 = 0 & signvv2 = 0 & signvv4 = 1
OR
signhv2 = 0 & signhv4 = 1 & signvv2 = 1 & signvv4 = 0
OR
signhv2 = 1 & signhv4 = 0 & signvv2 = 1 & signvv4 = 0
THEN slopeDisQual = 1
ELSE slopeDisQual = 0
436
hvect = hv2 + hv4
vvect = vv2 + vv4
450
If hvect < 0 AND vvect > 0
OR hvect > 0 AND vvect < 0
THEN Sign = 1
Else Sign = 0
438
abshvect = abs(hvect)
absvvect = abs(vvect)
If
abs(abshvect - absvvect)
is LESS THAN
(abshvect + absvvect)/diagDetThresh
THEN
DiagonalDetect = 1 (True)
ELSE
DiagonalDetect = 0 (False)
446
440
unQualSlopeFade =
abs(abshvect - absvvect)/diagthresh
448
If diagDetect = 1 AND slopeDisQual = 0
then
SlopeFade = unQualSlopeFade
else
SlopeFade = 0
diagDetect
slopeDisQual
Sign
414
416
slopeFade

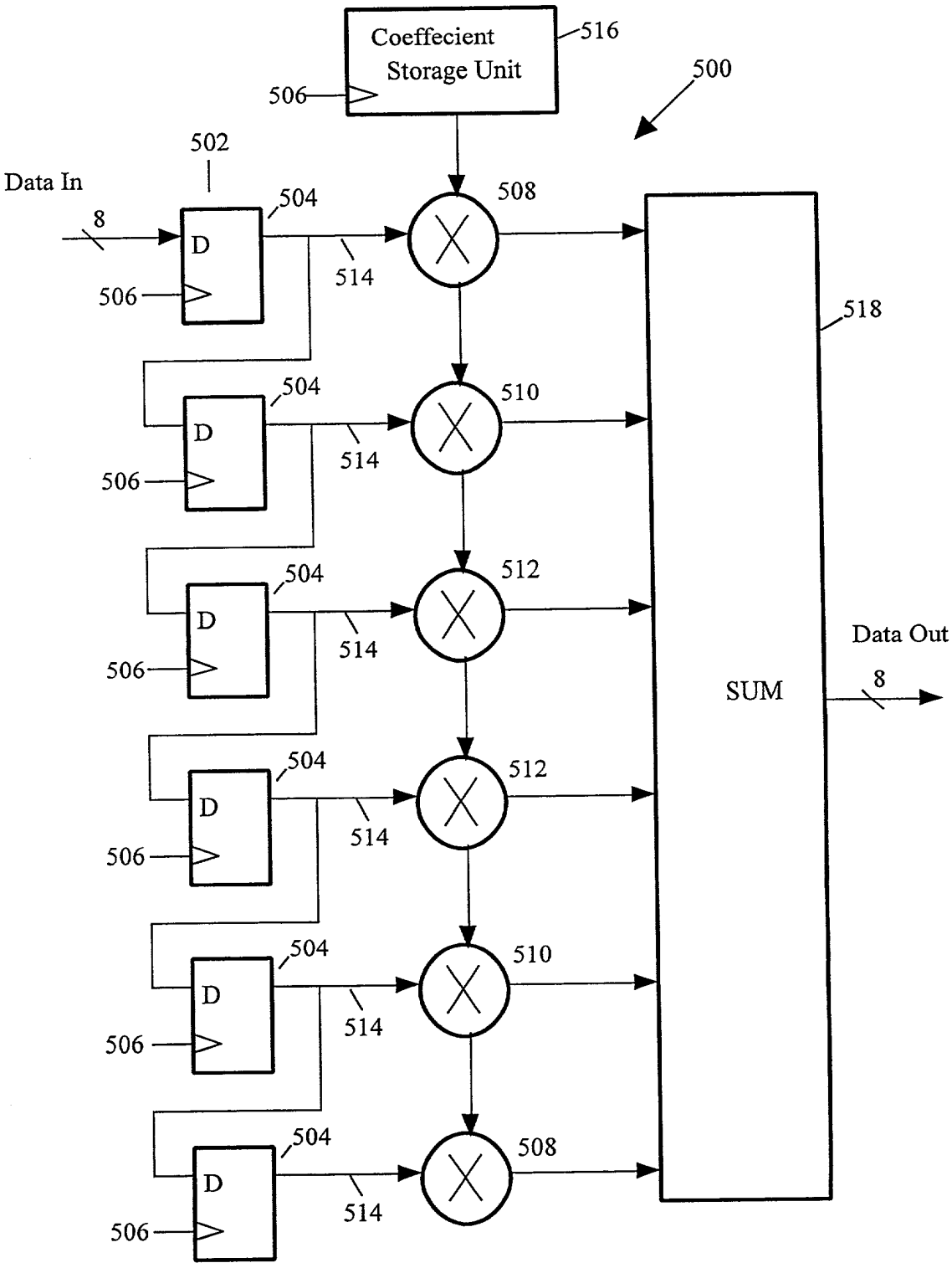

Coeffecient
Storage Unit
516
506
500
502
Data In
8
504
508
D
506
514
518
504
510
D
506
514
504
512
D
506
514
Data Out
SUM
8
504
512
D
506
514
504
510
D
506
514
504
508
D
506
514

Figure 22

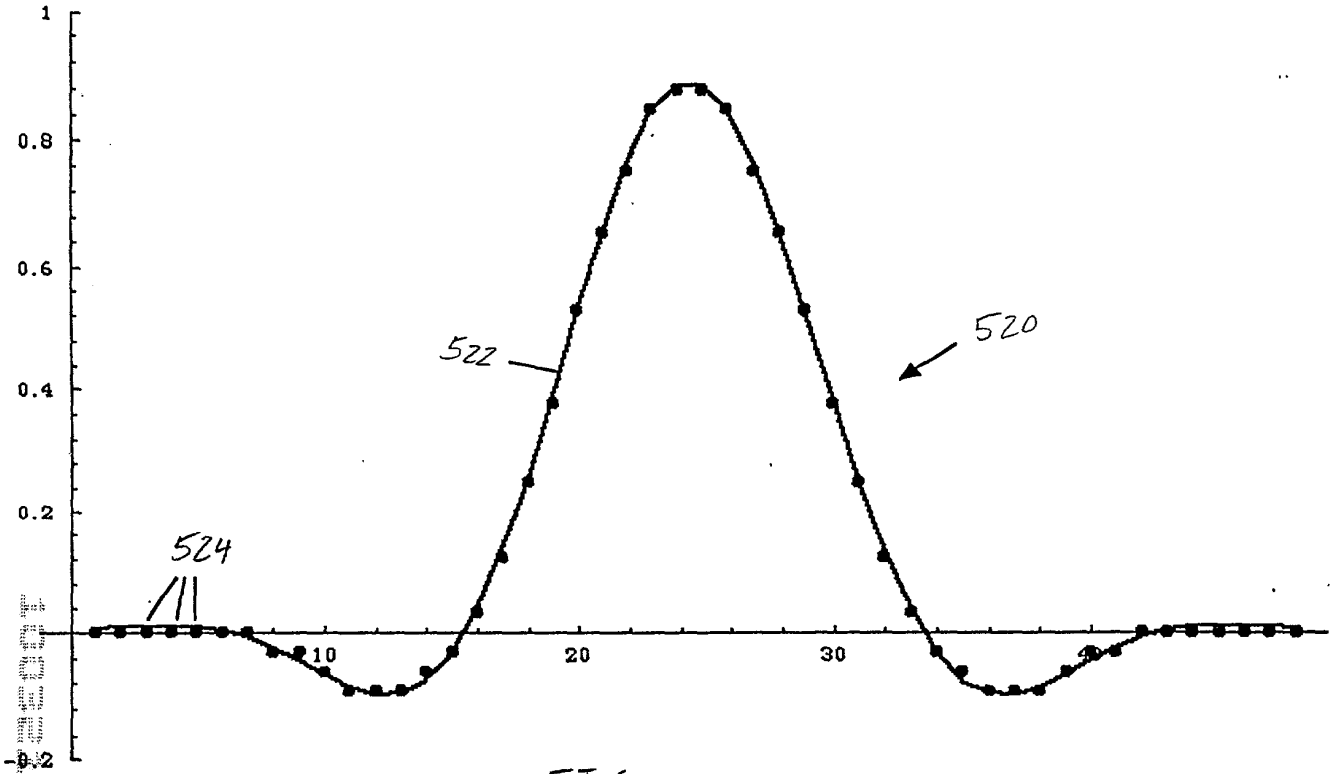


FIG. 23

set(1)	c(1), c(9), c(17), c(25), c(33), c(41)
set(2)	c(2), c(10), c(18), c(26), c(34), c(42)
set(3)	c(3), c(11), c(19), c(27), c(35), c(43)
set(4)	c(4), c(12), c(20), c(28), c(36), c(44)
set(5)	c(5), c(13), c(21), c(29), c(37), c(45)
set(6)	c(6), c(14), c(22), c(30), c(38), c(46)
set(7)	c(7), c(15), c(23), c(31), c(39), c(47)
set(8)	c(8), c(16), c(24), c(32), c(40), c(48)

FIG. 24

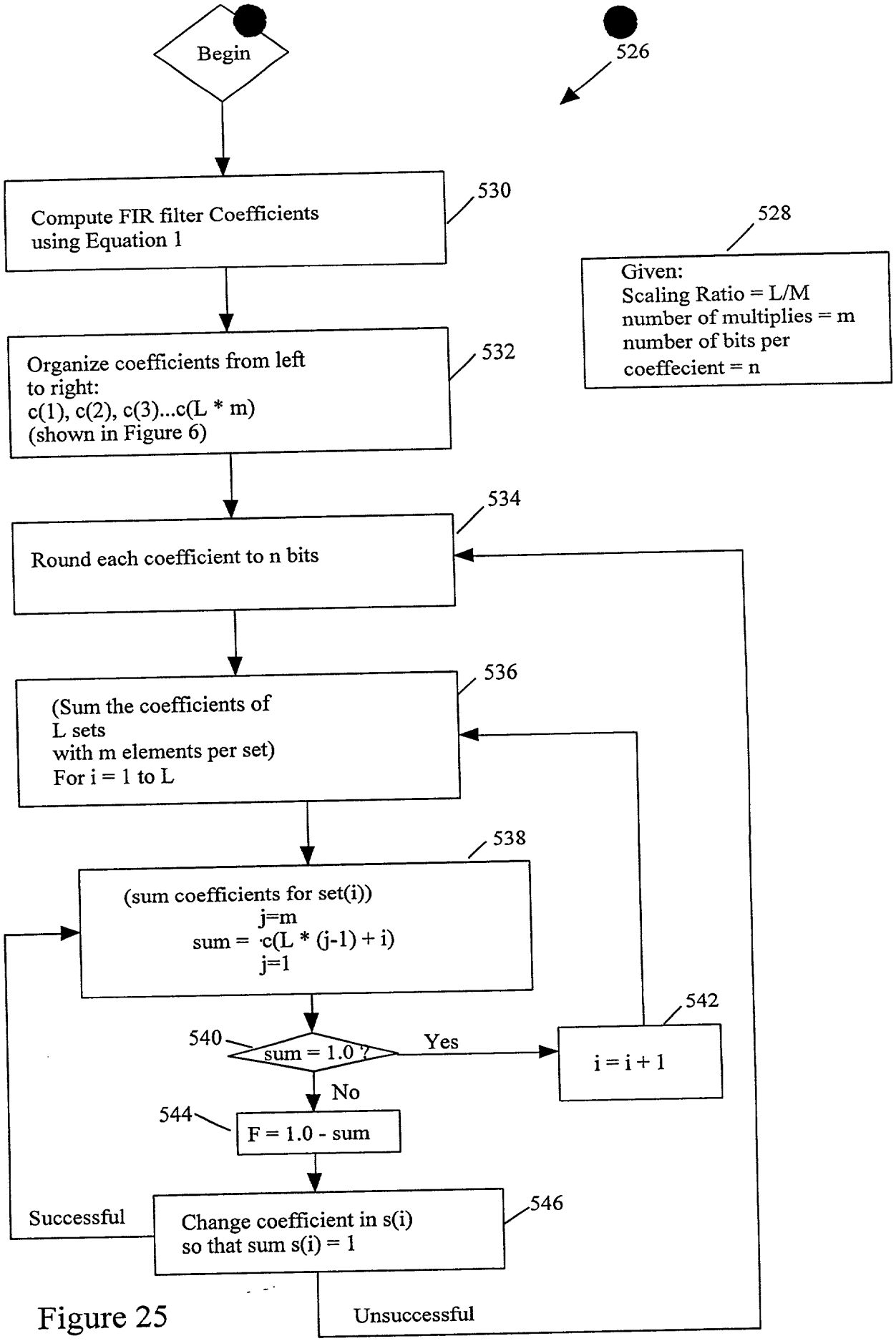

Begin
526
Compute FIR filter Coefficients using Equation 1
530
528
Given:
Scaling Ratio = L/M
number of multiplies = m
number of bits per coeffecient = n
Organize coefficients from left to right:
c(1), c(2), c(3)...c(L * m)
(shown in Figure 6)
532
Round each coefficient to n bits
534
(Sum the coefficients of
L sets
with m elements per set)
For i = 1 to L
536
(sum coefficients for set(i))
sum = c(L * (j-1) + i)
j=1 to j=m
538
sum = 1.0 ?
540
Yes
i = i + 1
542
No
F = 1.0 - sum
544
Change coefficient in s(i) so that sum s(i) = 1
546
Successful
Unsuccessful

Figure 25

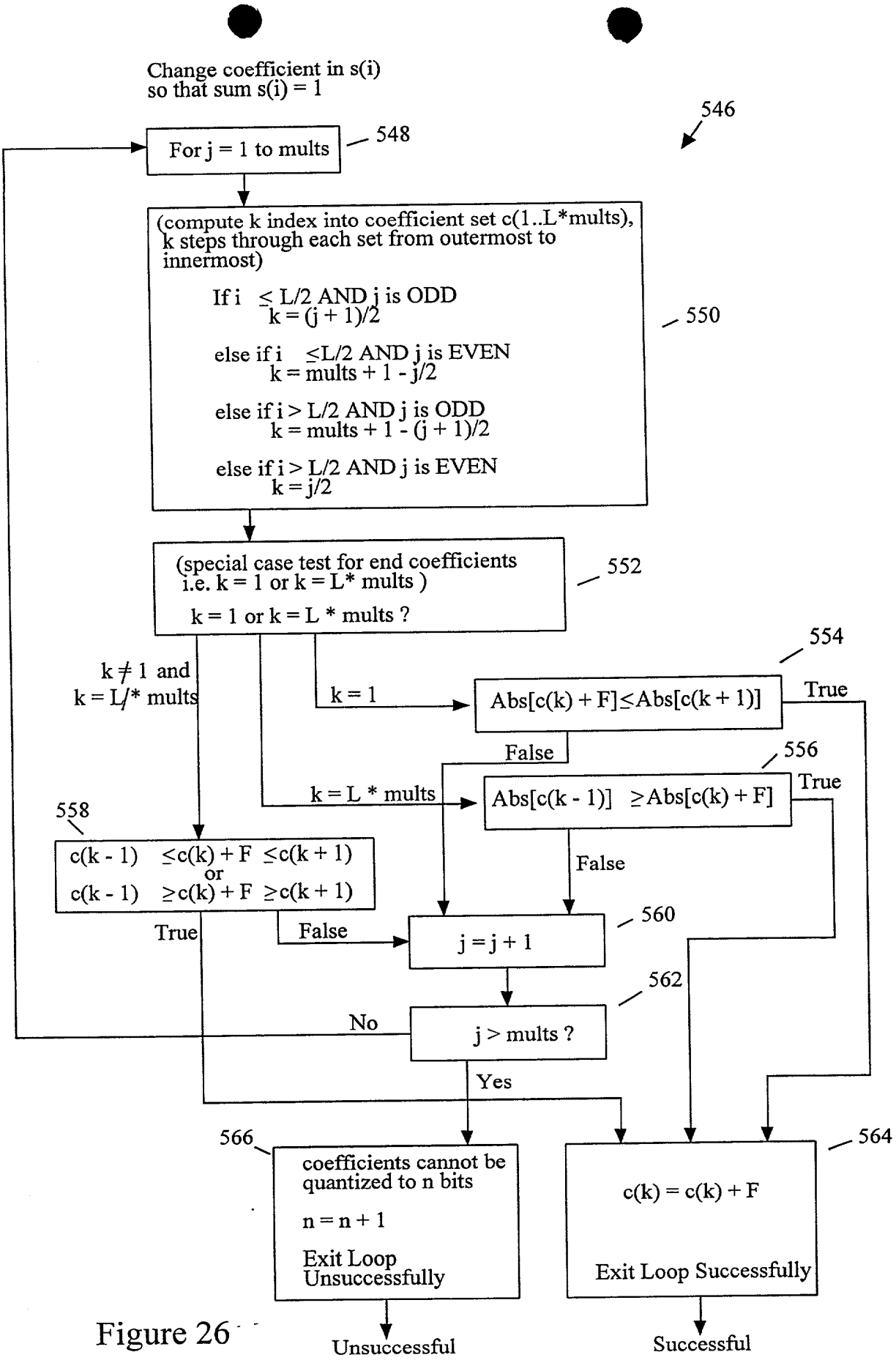

Change coefficient in s(i)
so that sum s(i) = 1
546
For j = 1 to mults
548
(compute k index into coefficient set c(1..L*mults),
k steps through each set from outermost to
innermost)
If i ≤ L/2 AND j is ODD
k = (j + 1)/2
else if i ≤L/2 AND j is EVEN
k = mults + 1 - j/2
else if i > L/2 AND j is ODD
k = mults + 1 - (j + 1)/2
else if i > L/2 AND j is EVEN
k = j/2
550
(special case test for end coefficients
i.e. k = 1 or k = L* mults)
k = 1 or k = L * mults ?
552
554
k ≠ 1 and
k = L/* mults
k = 1
Abs[c(k) + F]≤Abs[c(k + 1)]
True
False
556
k = L * mults
Abs[c(k - 1)] ≥Abs[c(k) + F]
True
False
558
c(k - 1) ≤c(k) + F ≤c(k + 1)
or
c(k - 1) ≥c(k) + F ≥c(k + 1)
True
False
560
j = j + 1
562
No
j > mults ?
Yes
566
coefficients cannot be
quantized to n bits
n = n + 1
Exit Loop
Unsuccessfully
564
c(k) = c(k) + F
Exit Loop Successfully
Unsuccessful
Successful

Figure 26

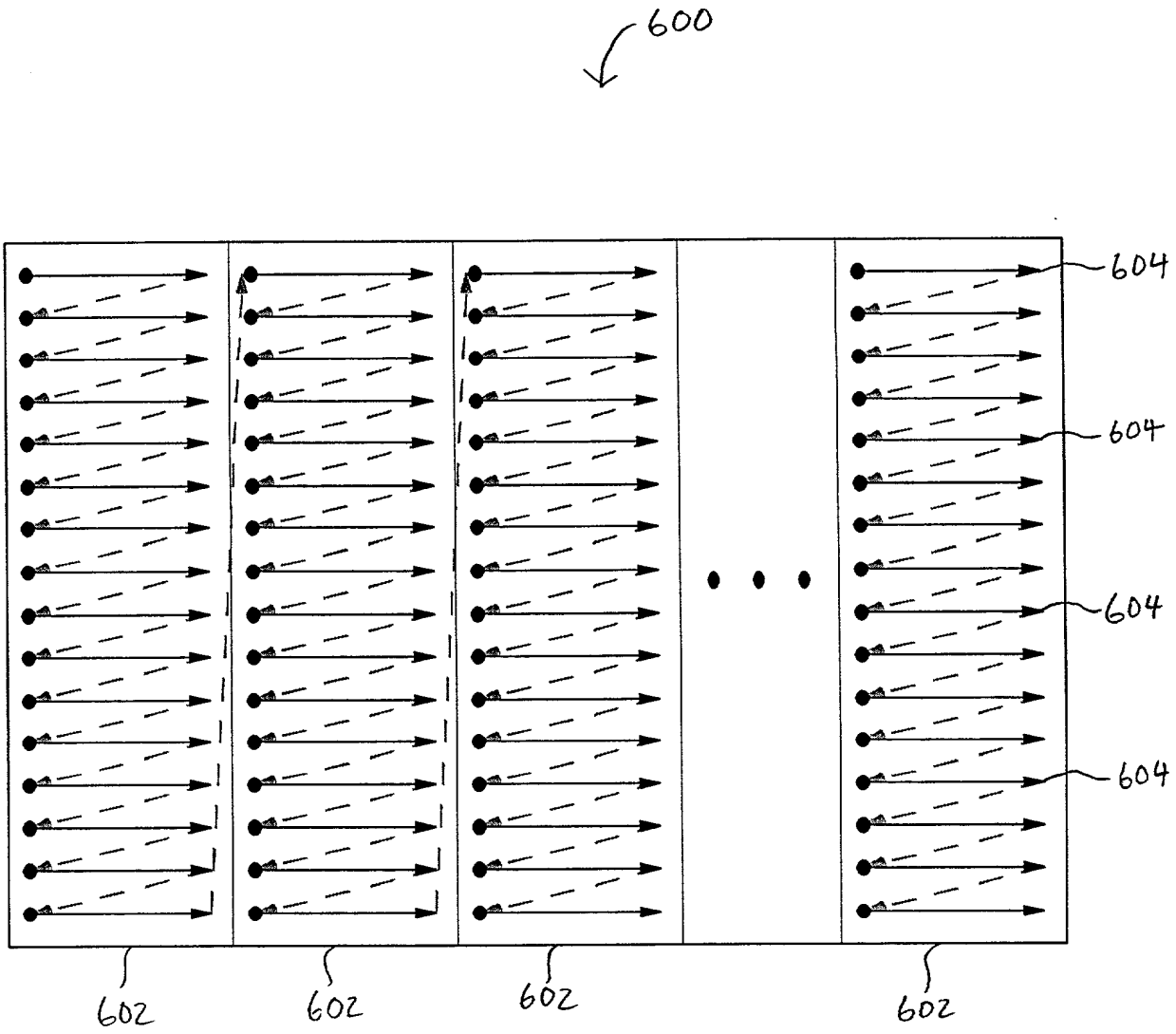


Figure 27

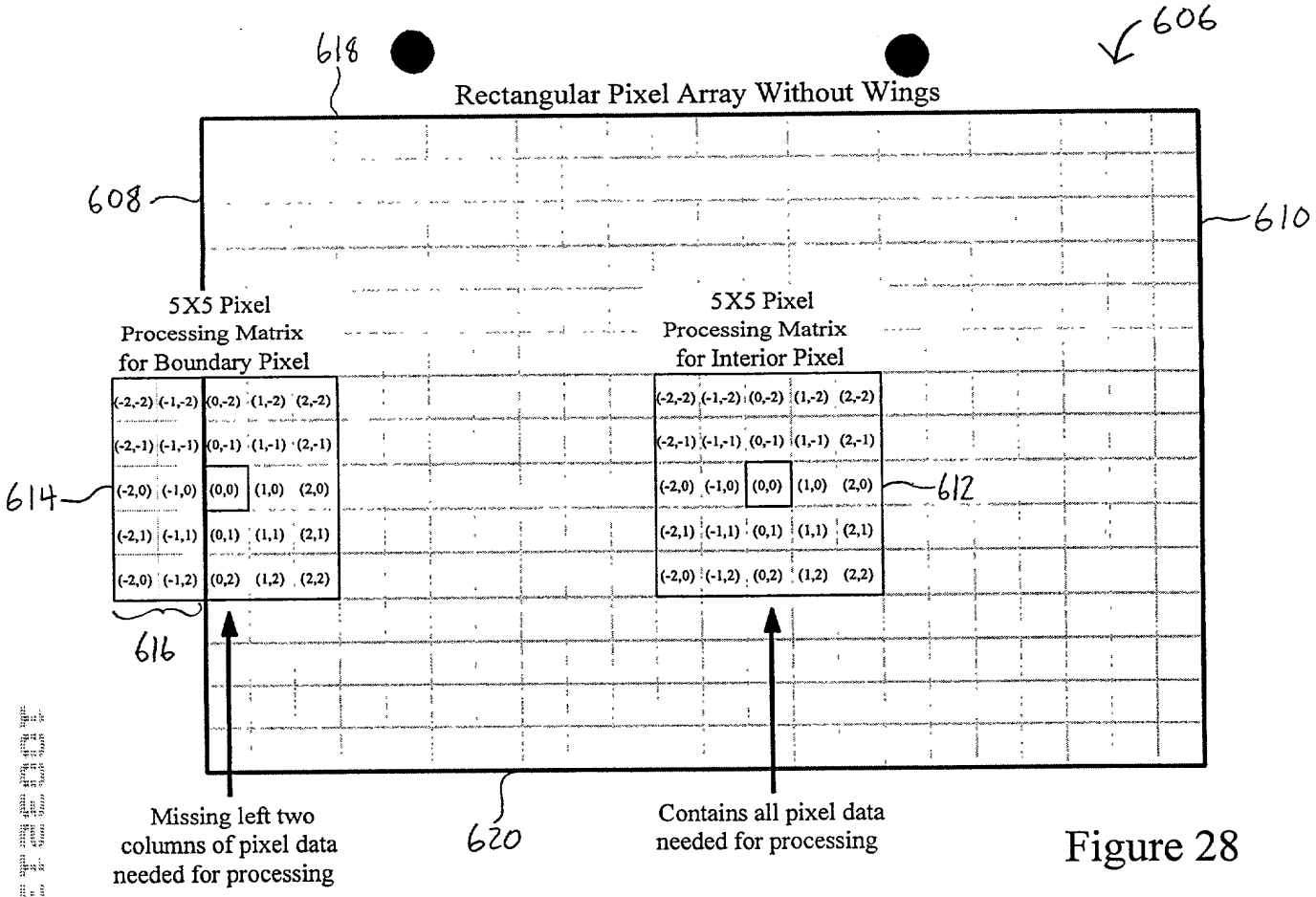


Figure 28

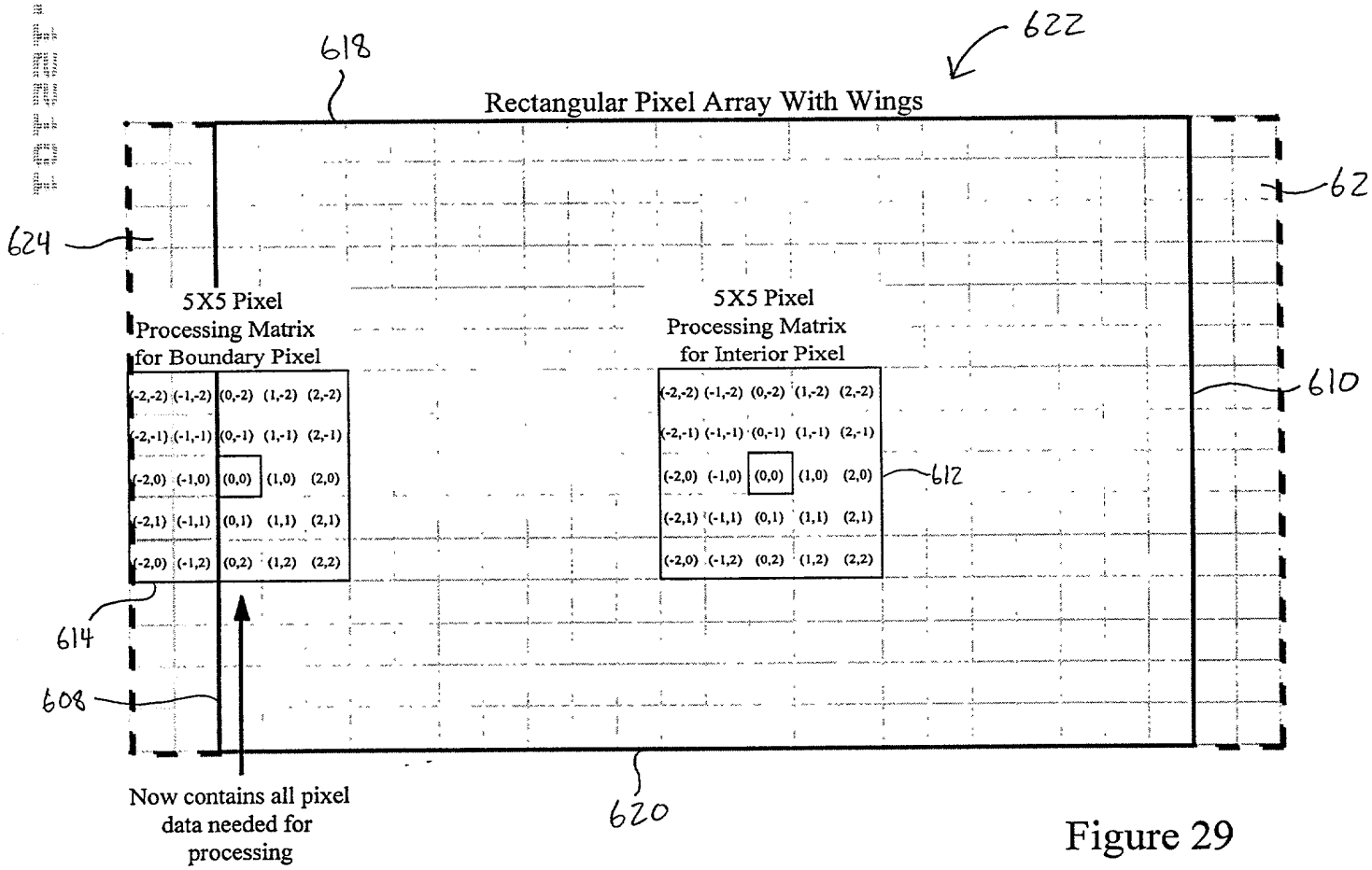


Figure 29

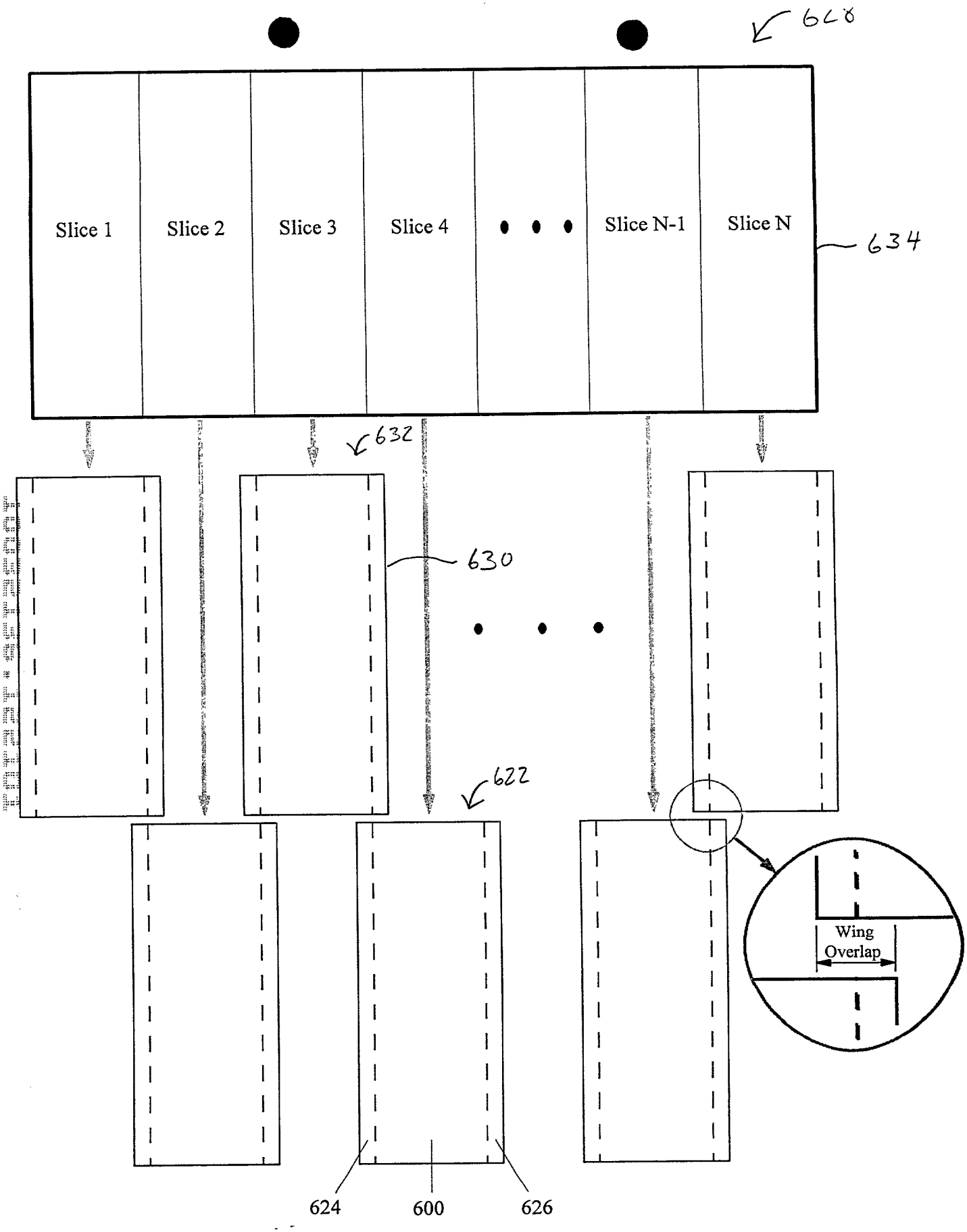

628
Slice 1
Slice 2
Slice 3
Slice 4
Slice N-1
Slice N
634
632
630
622
Wing
Overlap
624
600
626

Figure 30

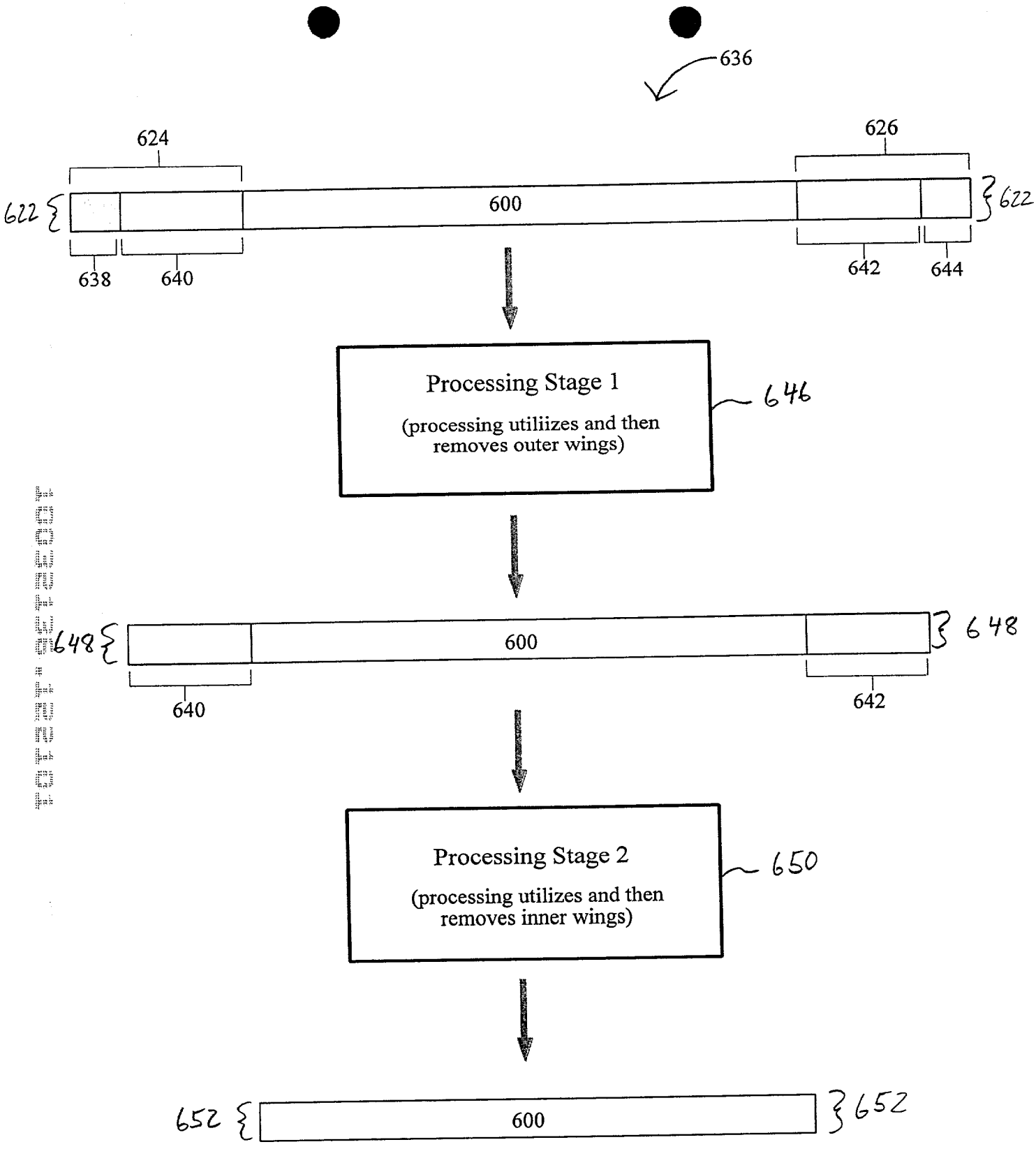

636
624
626
622
600
622
638
640
642
644
Processing Stage 1
(processing utiliizes and then removes outer wings)
646
648
600
648
640
642
Processing Stage 2
(processing utilizes and then removes inner wings)
650
652
600
652

Figure 31

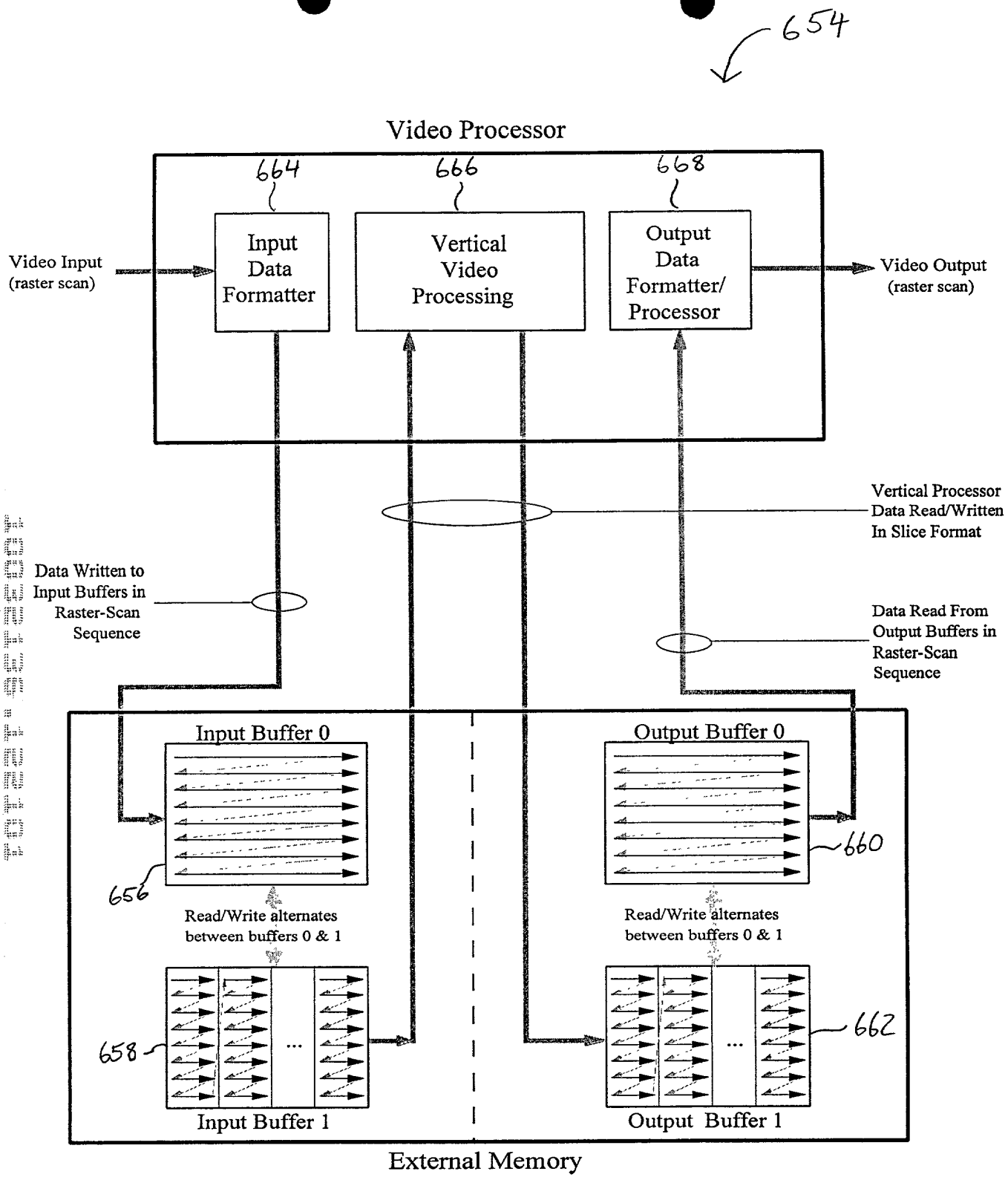

654
Video Processor
664
666
668
Input Data Formatter
Vertical Video Processing
Output Data Formatter/ Processor
Video Input (raster scan)
Video Output (raster scan)
Vertical Processor Data Read/Written In Slice Format
Data Written to Input Buffers in Raster-Scan Sequence
Data Read From Output Buffers in Raster-Scan Sequence
Input Buffer 0
Output Buffer 0
656
660
Read/Write alternates between buffers 0 & 1
Read/Write alternates between buffers 0 & 1
658
662
Input Buffer 1
Output Buffer 1
External Memory

Figure 32

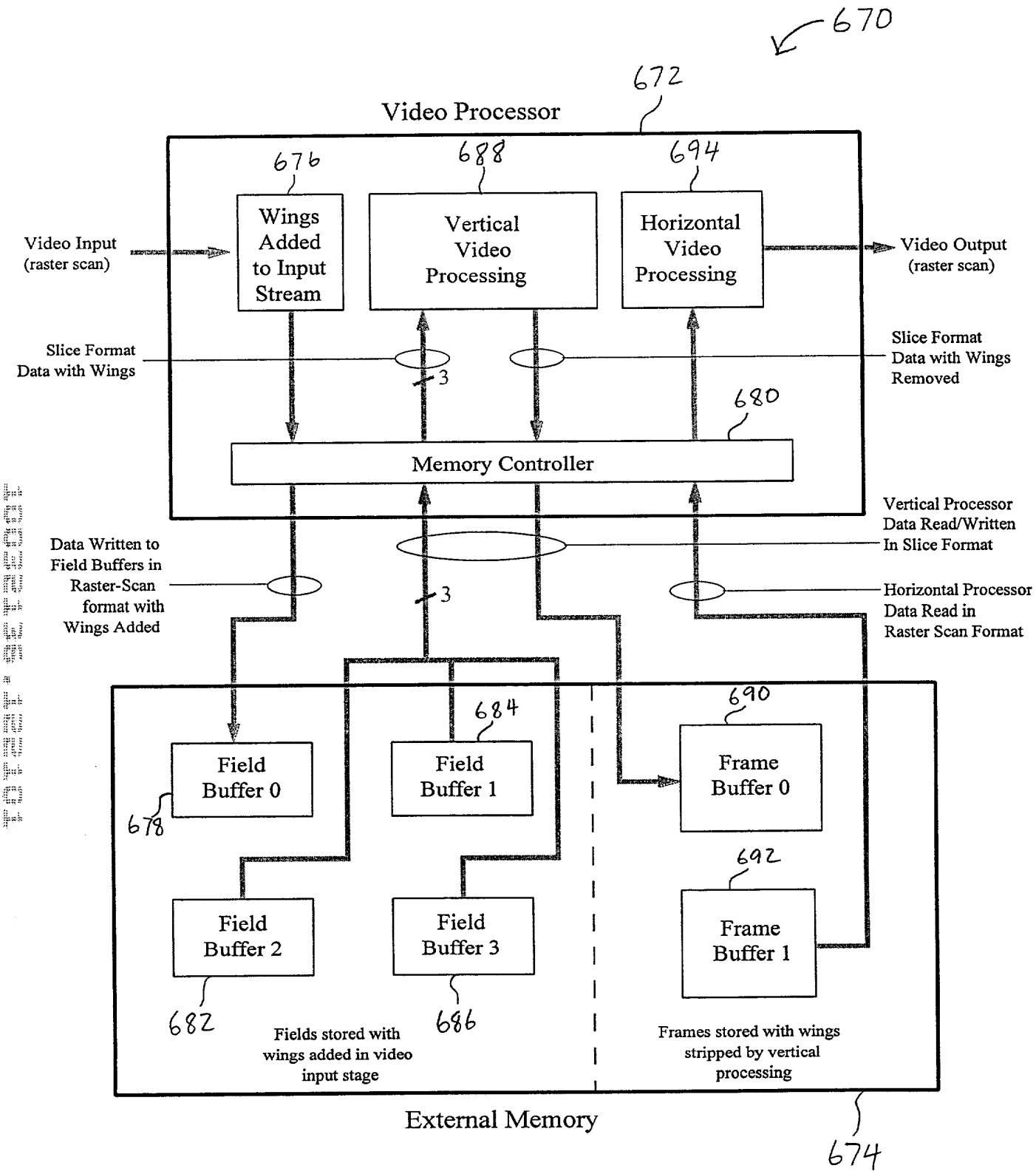

670
672
Video Processor
676
688
694
Wings Added to Input Stream
Vertical Video Processing
Horizontal Video Processing
Video Input (raster scan)
Video Output (raster scan)
Slice Format Data with Wings
Slice Format Data with Wings Removed
3
680
Memory Controller
Vertical Processor Data Read/Written In Slice Format
Data Written to Field Buffers in Raster-Scan format with Wings Added
3
Horizontal Processor Data Read in Raster Scan Format
690
684
Field Buffer 0
Field Buffer 1
Frame Buffer 0
678
692
Field Buffer 2
Field Buffer 3
Frame Buffer 1
682
686
Fields stored with wings added in video input stage
Frames stored with wings stripped by vertical processing
External Memory
674

Figure 33

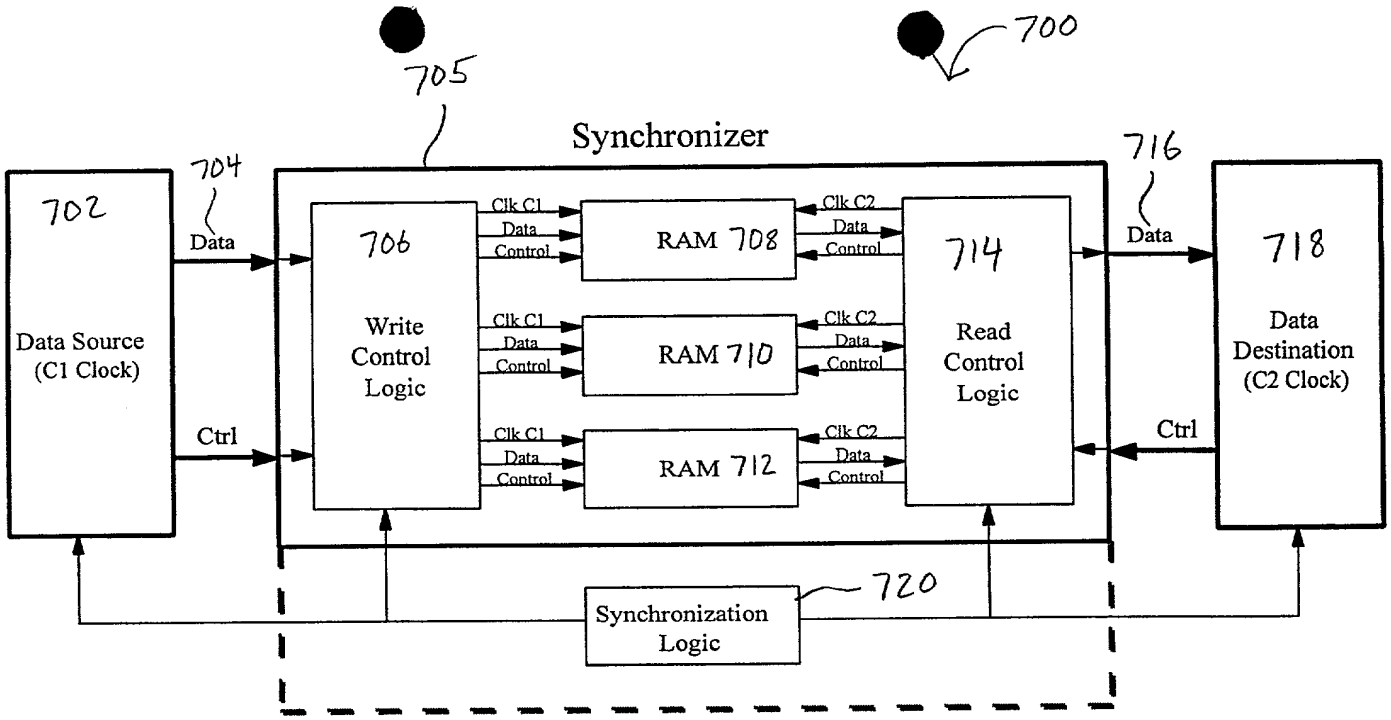


Figure 34

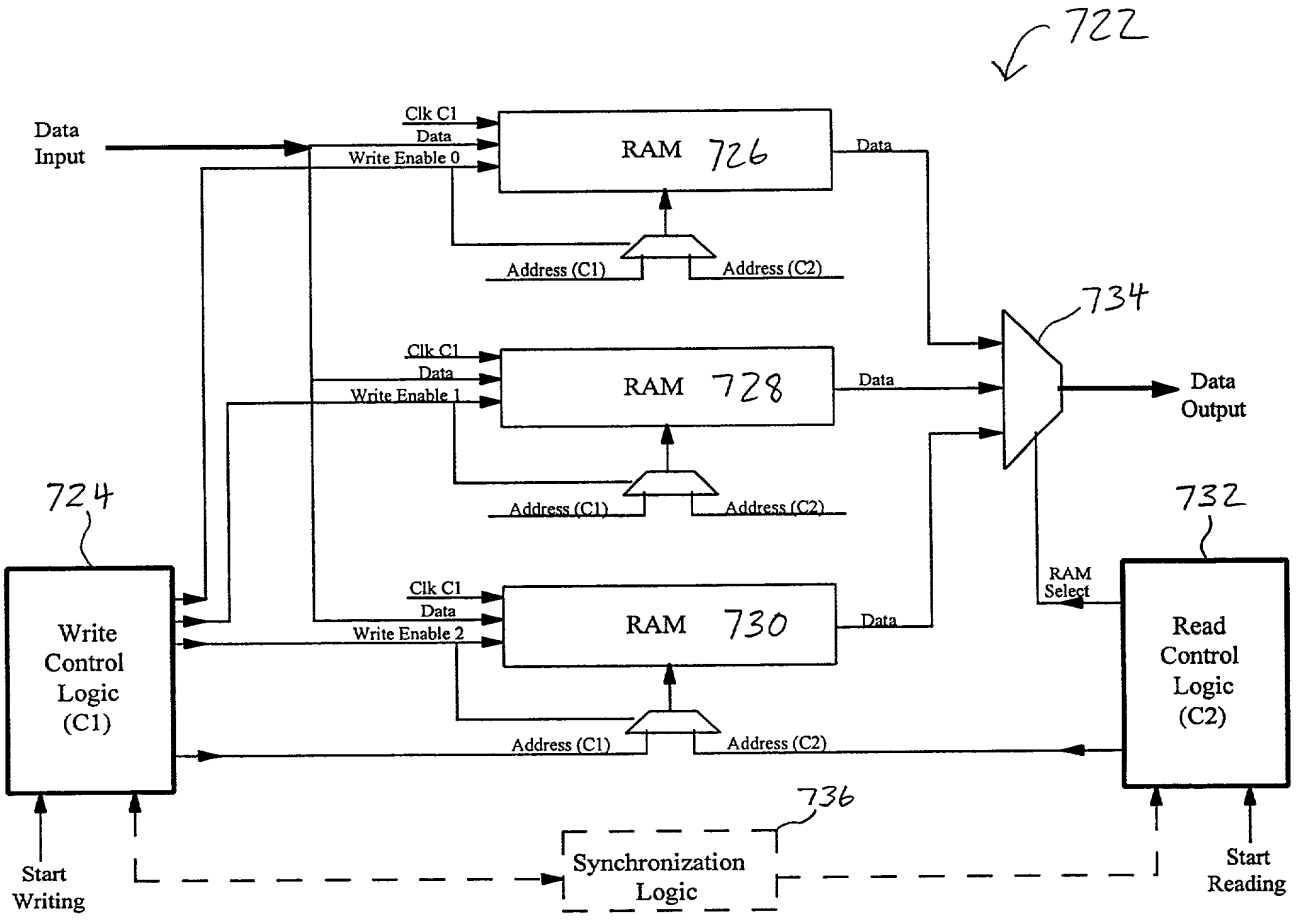
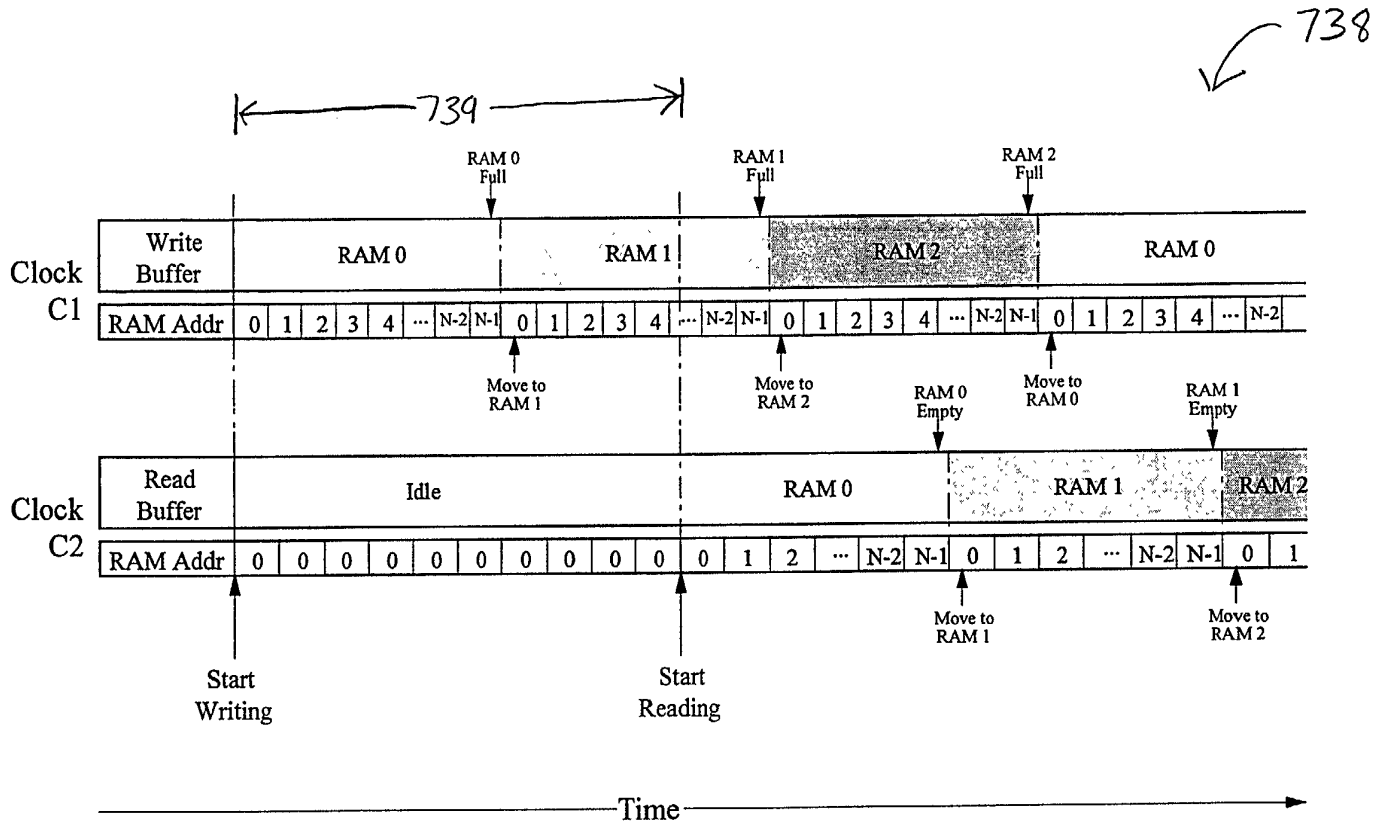
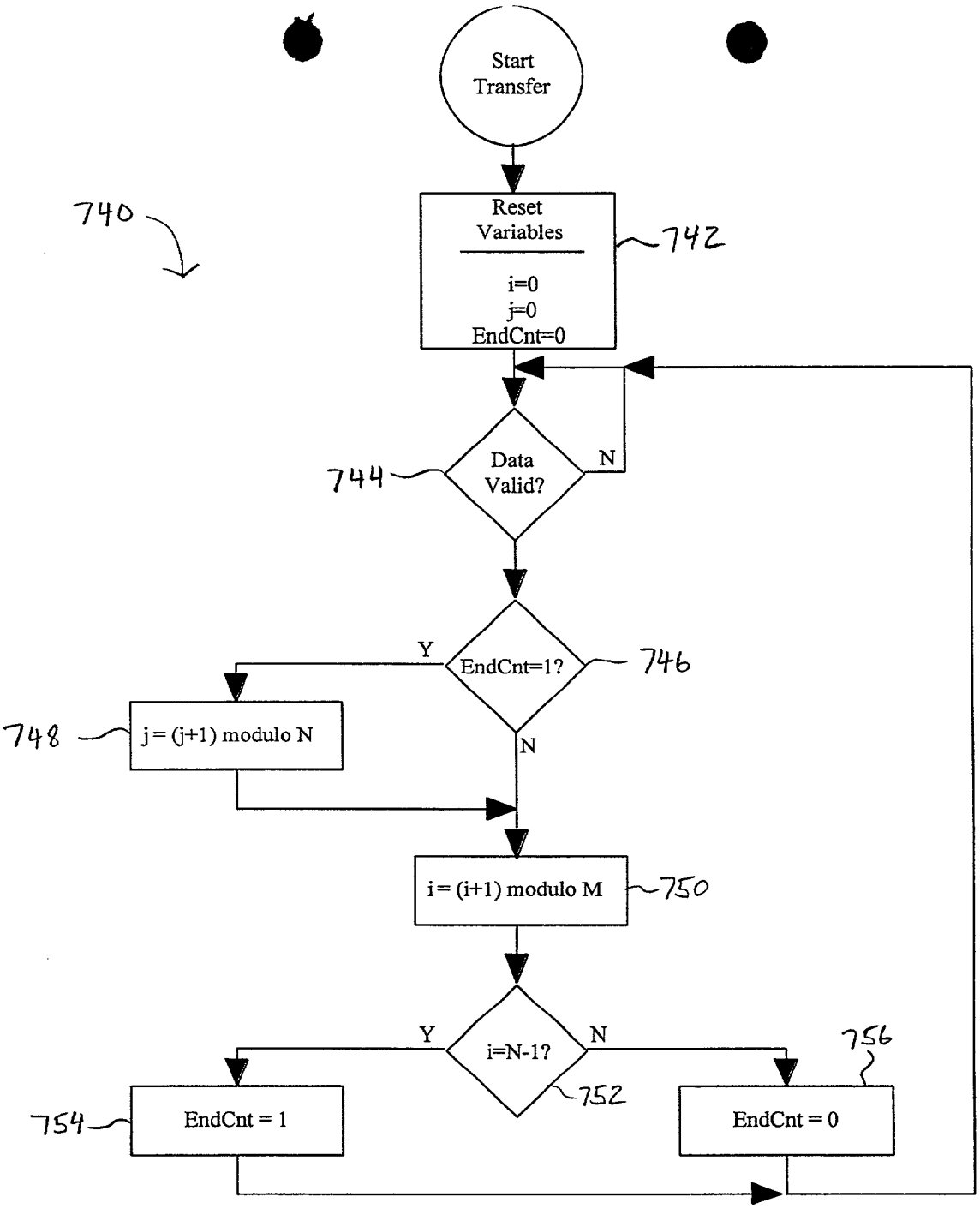


Figure 35



Note: Each RAM Has N Addressable Locations

Figure 36



NOTES:

1. "i" is the RAM Address
2. "j" denotes the selected RAM module
   the address MUX control and RAM write enable)
3. "EndCnt" indicates that the RAM Address points to the last location in a RAM module
4. "M" is the number of addressable location in a RAM module
5. "N" is the number of RAM modules

Figure 37

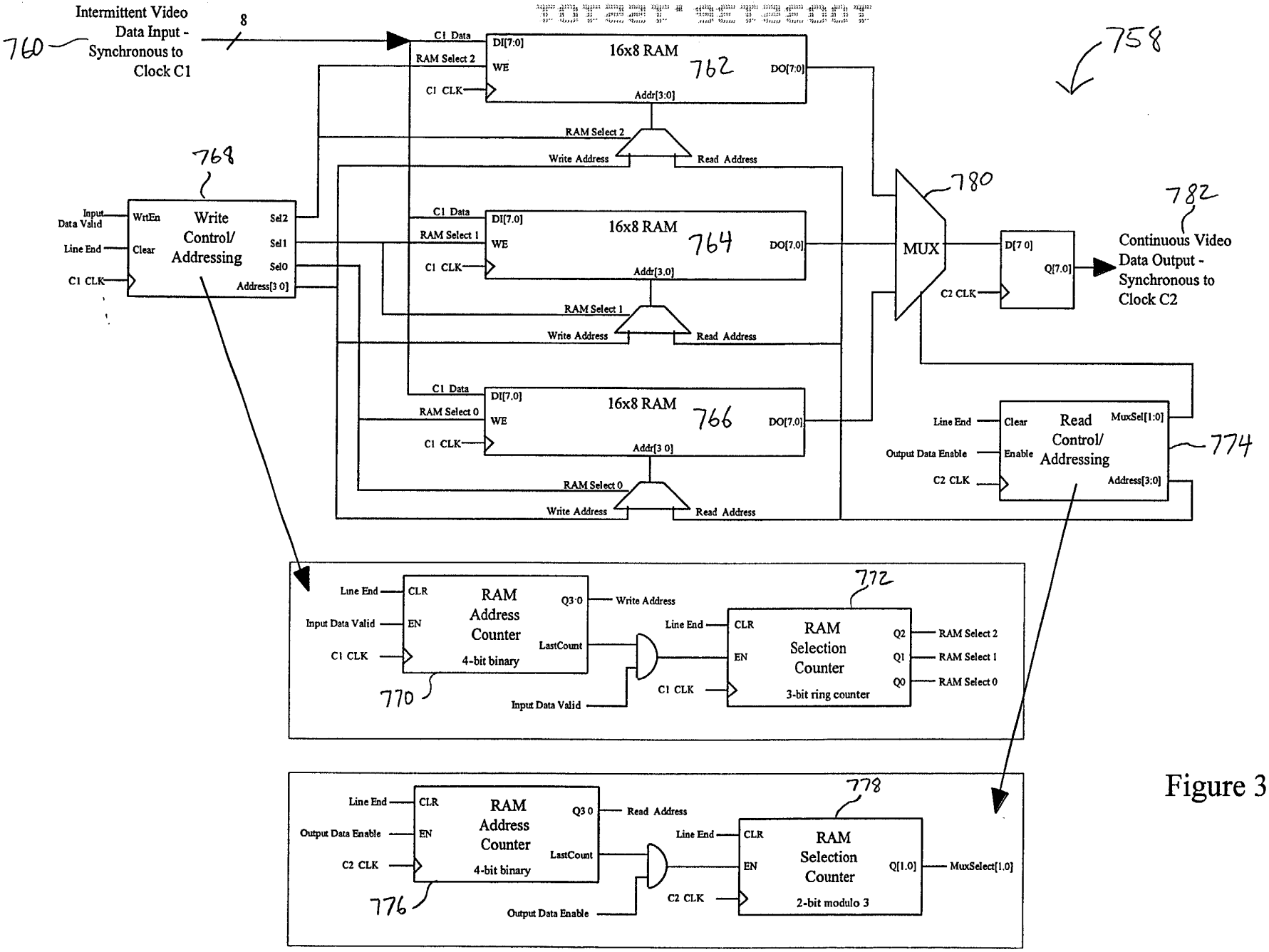

Intermittent Video Data Input - Synchronous to Clock C1
760
8
758
C1 Data
RAM Select 2
C1 CLK
DI[7:0]
WE
16x8 RAM
762
DO[7:0]
Addr[3:0]
RAM Select 2
Write Address
Read Address
768
Input Data Valid
Line End
C1 CLK
WrtEn
Clear
Write Control/ Addressing
Sel2
Sel1
Sel0
Address[3 0]
C1 Data
RAM Select 1
C1 CLK
DI[7.0]
WE
16x8 RAM
764
DO[7.0]
Addr[3.0]
RAM Select 1
Write Address
Read Address
780
MUX
782
D[7 0]
Q[7.0]
C2 CLK
Continuous Video Data Output - Synchronous to Clock C2
C1 Data
RAM Select 0
C1 CLK
DI[7.0]
WE
16x8 RAM
766
DO[7.0]
Addr[3 0]
RAM Select 0
Write Address
Read Address
Line End
Output Data Enable
C2 CLK
Clear
Enable
Read Control/ Addressing
MuxSel[1:0]
Address[3:0]
774
Line End
Input Data Valid
C1 CLK
CLR
EN
RAM Address Counter
4-bit binary
Q3:0
LastCount
Write Address
770
772
Line End
Input Data Valid
C1 CLK
CLR
EN
RAM Selection Counter
3-bit ring counter
Q2
Q1
Q0
RAM Select 2
RAM Select 1
RAM Select 0
Line End
Output Data Enable
C2 CLK
CLR
EN
RAM Address Counter
4-bit binary
Q3 0
LastCount
Read Address
776
778
Line End
Output Data Enable
C2 CLK
CLR
EN
RAM Selection Counter
2-bit modulo 3
Q[1.0]
MuxSelect[1.0]

Figure 38